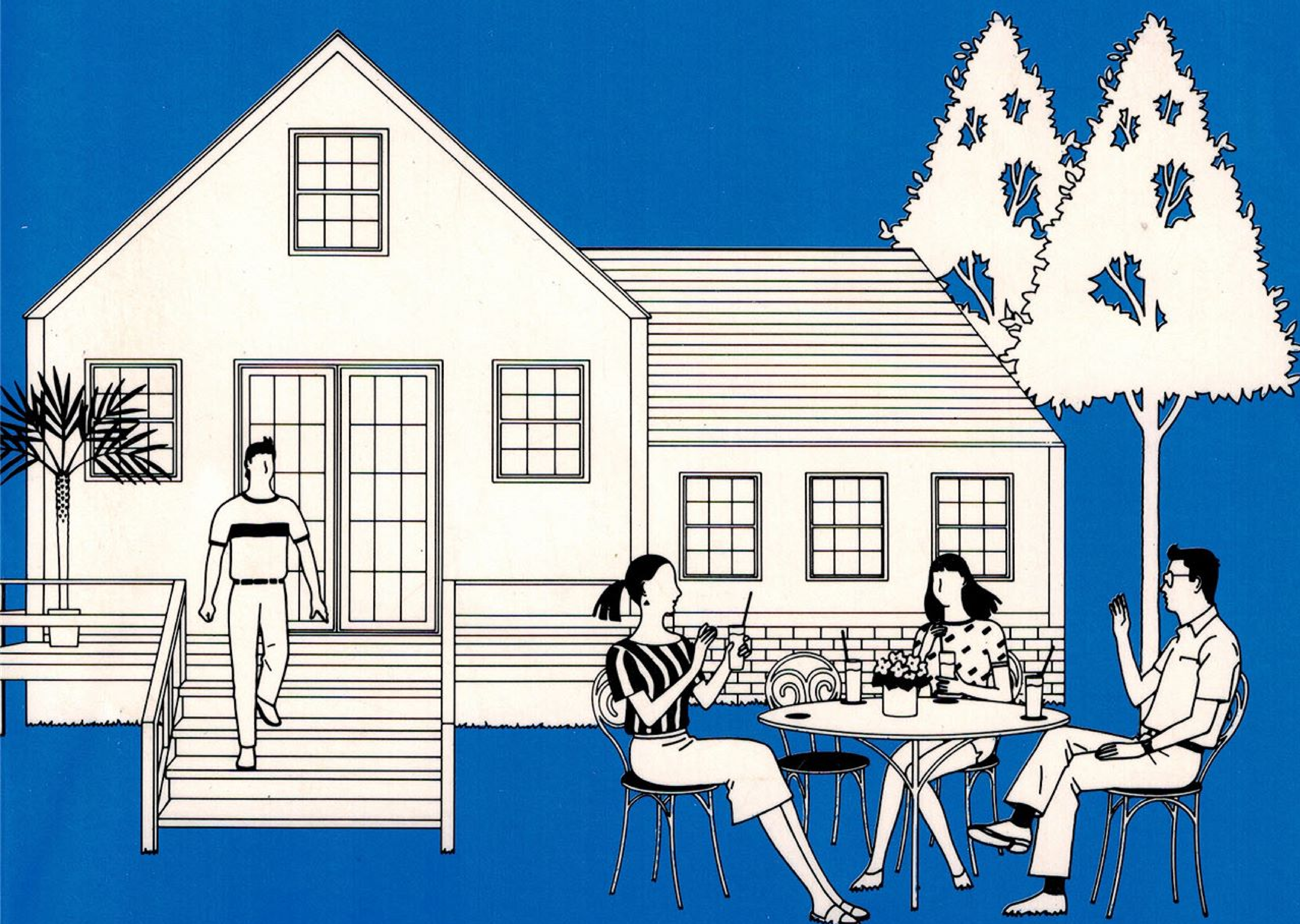
NECパーソナルコンピュータ
PC-9800シリーズ

NEC

PC-9801UV21

BASICユーザーズマニュアル



PC-9801UV21-BU

御注意

(1) 本書の内容の一部または全部を無断転載することは禁止されています.

(2) 本書の内容に関しては将来予告なしに変更することがあります.

(3) 本書は内容について万全を期して作成いたしましたが, 万一御不審な点や誤り, 記載もれなどお気付きのことがありましたら, お買い求めの販売店または最寄りのBit-INN へ御連絡ください.

(4) 運用した結果の影響については(3)項にかかわらず責任を負いかねますので御了承ください.

PC-9801UV21

BASIC ユーザーズ マニュアル

まえがき

本書は本体に標準装備されているN88-BASIC(86) システムの機能について詳細に解説した説明書です.

N88-BASIC(86) の言語機能（言語の規則，個々の命令の機能等）については「BASICリファレンスマニュアル」を御参照下さい.

また，ハードウエアの機能については「ハードウエアマニュアル」を御参照ください.

目　次

このマニュアルの構成

本書はN_{88}-日本語BASIC(86)(Ver 4.0)の機能を中心に解説した説明書です.

第1章および第2章においては，N_{88}-BASIC(86) システムの種類とその起動方法について解説しています.

第3章および第4章においては，キーボードの基本的な操作およびN_{88}-BASIC(86) システムのスクリーンエディタについて解説しています.

第5章および第6章においては，N_{88}-BASIC(86) システムの日本語処理機能，すなわち，日本語入力の方法および日本語の操作等について解説しています.

第7章，第8章および第9章においては，ディスプレイ画面の制御，すなわちテキスト画面に対する文字の表示，グラフィックス機能あるいは画面ハードコピー機能等について解説しています.

第10章，第11章および第12章においては，ファイル制御，特にディスクファイルの取り扱いについて解説しています.

第13章および第14章においては，N_{88}-BASIC(86) システムの特殊モードであるターミナルモードおよびモニタモードについて解説しています.

第15章においては，システムディスクに格納されている各種のユーティリティプログラムに関して解説しています.

第16章においては，マウスを制御するためのソフトウエアドライバの機能および使用方法について解説しています.

また第17章ではライトペンの制御方法について解説しています.

第18章においては，N_{88}-BASIC(86) システムにおける機械語プログラムの作り方について解説しています.

第19章では，拡張機能について解説しています.

第20章および第21章では各種のスイッチ（ディップスイッチ，メモリスイッチ）について解説しています.

第22章，第23章においては，各種メモリ構成について解説しています.

第1章

N_{88}-BASIC(86)システムの起動

PC-9800シリーズ各機種は本体に96KBのROMを内蔵しています.

このROMの中には次に示すようなN_{88}-BASIC(86)システムの基本機能が組み込まれています.

・スクリーンエディタ（JISコード入力方式による日本語入力も使用できます）

・ディスク装置を除く各種入出力機器（ディスプレイ，キーボード，プリンタ，RS-232C，カセットテープ等）の制御機能

・グラフィック制御機能

・各種の演算機能

・単精度数値データのための関数（SIN, COS, TAN等の関数）

・ターミナルモード，モニタモード等の特殊機能

従って，このROMに組み込まれた基本機能だけを使用して，プログラムの入力およびプログラムの実行がある程度可能です.

フロッピィディスク装置にシステムディスクをセットしないで，電源をONにしますと（すでに電源がONになっている場合，リセットスイッチを押します），ROMに組み込まれた基本機能だけを持つシステムが立ち上がります．この環境を**ROM BASICモード**と呼びます.
また，この時のBASICシステムを**ROMモードBASIC**と呼びます.

ROM BASICモードにおいてはディスク装置が使用できませんので，プログラムはスクリーンエディタを使用し，キーボードからいちいち入力していかねばなりませんし，入力したプログラムを保存することもできません．カセットテープレコーダが使用できますが，ディスク装置に比べ記憶容量あるいは操作性の面で劣ります.

フロッピィディスク装置を使用すれば操作性が飛躍的に向上します.

本体の電源をONにした後（すでに電源がONになっている場合，リセットスイッチを押します)，システムディスクをフロッピィディスク装置にセットしますと（フロッピィディスク装置のどのドライブを使用してもかまいません)，ROMモードBASICの機能に，次に示す機能が付加された形でシステムが立ち上がります.

・ディスク装置の制御機能

・倍精度数値データのための関数（SIN,COS,TAN等の関数）

・日本語文字列操作のための文や関数

・DRAW文（複雑な図形や描画を行うのに便利な機能です）

・連文節変換入力方式あるいは単文節変換入力方式による日本語入力機能*

・拡張グラフィック機能（中間色の表示や高速グラフィック描画が可能になります)*

・モデムNCU内蔵電話制御機能*

・拡張モニタ機能*

・拡張画面ハードコピー機能*

この環境をDISK BASICモードと呼びます.

また，この時のBASICシステムをDISKモードBASICと呼びます.

備　考　*印の付いた機能はその使用の有無が選択可能です.

詳しくは「第2章　DISKモードBASIC使用時の種々の機能選択」を参照して下さい.

DISK BASICモードにおいてはこれらの付加機能に加え，システムディスクに格納されている

各種ユーティリティプログラム（ディスク装置関連のものが主です）

マウス用ソフトウェアドライバ（マウスを制御するためのプログラムです）

等を使用することができます.

1.1　ROMモードBASICの立ち上げ

(1)　本体にフロッピィディスク装置が内蔵されている場合，2台とも空（フロッピィディスクがセットされていない状態）であることを確認します.

(2)　本体に外付型のディスク装置（フロッピィディスク装置あるいは固定ディスク装置）がある場合，装置の電源をOFFにします.

(3)　本体の電源をONにして下さい（すでに電源がONの状態にある時はリセットスイッチを押します）.

10〜15秒後にROMモードBASICが立ち上がり次のメッセージがディスプレイ装置に表示されます.

How many files (0-15) ? ■

このメッセージは「いくつのファイルを使用しますか」と尋ねているものです.

ROM BASICモードでは，一般的にファイルは使用しませんので，[↵]キーを入力します.

ディスプレイ画面が次の状態となります.

以降，ROMモードBASICの世界です.

```
How many files (0-15)?
NEC N-88 BASIC(86) version 2.0
Copyright (C) 1983 by NEC Corporation/Microsoft Corp.
××××××Bytes free
Ok
■
```

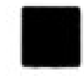
注　意　××××××は使用できるメモリ量を示しています.

注　意　ディップスイッチSW2の1は常にOFFの状態にしておいてください.

（このスイッチは出荷時からOFFの状態になっています）

また，SW2の2も通常はOFFにしておきます．このスイッチがONになっている場合，次のようなメッセージが表示されます.

```
How many files (0-15) ?
NEC N-88 BASIC(86) version 2.0
Copyright(C) 1983 by NEC Corporation/Microsoft Corp.
```

××××××Bytes free
Terminal mode


これはシステムがターミナルモードになっていることを示します．
ターミナルモードを使用しない場合，SW2の2をOFFにしてからリセットスイッチを押してください．ターミナルモードの詳細については「第13章　ターミナルモード」を参照してください．

注　意　××××××Bytes freeまで画面に表示され，OKが表示されない場合，次の原因が考えられます．

・拡張インタフェースボード（GP-IBボード，RS-232C（第2回線/第3回線）等のボードを指します）が実装されていないにもかかわらずメモリスイッチSW4のいずれかのビットがONになっている．

この場合，メモリスイッチSW4の該当ビットをOFFにしてください．

1.2　DISKモードBASICの立上げ

システムディスクには，文節変換入力用辞書ファイル“BUNSET.SU”が格納されています．
システムディスクを標準の状態で使用し，DISKモードBASICを立ち上げますと，連文節変換入力方式による日本語入力が行えます．
また，システムディスクは単文節変換入力用のシステムディスクに変更することができます．
日本語入力方法の変更は，“setup.n88”ユーティリティでおこないます（“setup.n88”ユーティリティについては，「第15章　システムディスクに格納されているユーティリティプログラム」を参照してください．また，「第2章　2.2　日本語入力方法の選択」にも解説があります）．
このシステムディスクを使用しDISKモードBASICを立上げますと単文節変換入力方式による日本語入力が行えます．
DISKモードBASICは一般にシステムディスクから立上げますが，固定ディスク装置が接続されている場合，システムディスクを使用せず，直接固定ディスク装置からDISKモードBASICを立ち上げることもできます．ただし，この場合，あらかじめ準備しておくことがあります．
「1.2.2　固定ディスクからの立上げ」を参照してください．

1.2.1　システムディスクからの立上げ

(1)　固定ディスク装置が接続されていない場合
システムディスクを次のタイミングでフロッピィディスク装置にセットします．

①　電源投入によりシステムを立ち上げる場合
電源を「ON」にした後，すぐにフロッピィディスク装置にシステムディスクをセットしなければなりません（4～5秒くらいの間にセットします）

②　すでに電源「ON」の状態にあり，リセットスイッチを押すことによりシステムを立ち上げる場合
システムディスクをセットした後，リセットスイッチを押してください．

システムディスクは最大4ドライブ（本体にフロッピィディスク装置が内蔵されている場合，内

蔵2ドライブ,増設2ドライブ,また,フロッピィディスク装置が内蔵されていない場合,外付け4ドライブ）の内のいずれのドライブにもセットできますが，ドライブ番号＃1以外のドライブにセットする場合は，それより小さい番号のドライブには，他のフロッピィディスク媒体をいっさいセットしないようにしてください（本体にフロッピィディスク装置が内蔵されている場合，通常，内蔵ドライブ＃1にシステムディスクをセットします).

注意1 電源投入によりシステムを立ち上げる場合，システムディスクを先にセットした後，電源を「ON」にしますと，システムディスクが，こわされるおそれがあります.

注意2 電源投入により，システムを立ち上げる場合，電源を「ON」にした後，約10秒以内に，システムディスクをセットしないと，通常，ROMモードBASICが立ち上がってしまいます.

　またこのとき，他のフロッピィディスク装置に，システムディスク以外のフロッピィディスクがセットされているとキーボードから何も入力できなくなるなどの誤動作を起こします.

③ 数秒たつと，システムディスクをセットしたドライブのアクセス表示用LEDが点灯します．これは，システムDISK CODEの読み込みが始まったことを示しています.

④ さらに数秒たつと，LEDが消え，同時にディスプレイにメッセージが現われます.

```
Disk version
How many files (0-15) ?■
```

⑤ ⏎ キーを入力します．(How many files (0-15) ?の詳細については，「10.3 ファイルの同時オープン数」を参照してください)

⑥ 次のようなメッセージが表示され，これでBASICの命令が入力可能となります.

```
Disk version
How many files (0-15) ?
NEC N-88 BASIC (86) version 4.0
Copyright (C) 1983 by NEC Corporation/Microstft Corp.
XXXXXX Bytes free
Ok
```



以降がDISKモードBASICの世界です.

(2) 固定ディスク装置が接続されている場合

固定ディスク装置が接続されている場合，固定ディスク使用のための操作が必要となります.

① How many files (0-15) ?■ までの操作は固定ディスク装置が接続されていない場合と同様です.

　ここで ⏎ キーを応答すると次のメッセージが表示されます.

② User identifier?■

この問い合わせに対して，これから使用するユーザ識別名を宣言します.

　ユーザ識別名は固定ディスクを複数の利用者で分割使用するための識別子です（たとえば，ユーザ識別名が異なれば同じファイル名のファイルを作成することができます．また，他の利用者から自分のファイルを保護することもできます).

　ユーザ識別名は英文字で始まる3桁以下の文字列によって定義されます． ⏎ キーで応答

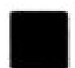
すると，ユーザー識別名は“△△△”（△はスペース（空白））となります．また，ユーザー識別名として，「共通ユーザ識別名」があります．「共通ユーザ識別名」は“999”として，システム固有なものとして定義されています．“999”で応答すると，これによって立ち上がったシステムの下で，登録されたファイルはすべて共通ユーザ識別名をもち，どのユーザからも使用することができるようになります．

ユーザ識別名は固定ディスク上のシステムを対象としているものです．フロッピィディスク上のファイルには関係ありません．

ユーザ識別名に関する詳細は「11.1.2　固定ディスクユニット固有機能の準備」を参照してください．

③ ユーザ識別名に“ABC”と応答したとします．後はシステムディスクから起動した場合と同じです．

```
Disk version
How many files (0-15)? 3
User identifier? ABC
NEC N-88 BASIC (86) version 4.0
Copyright (C) 1983 by NEC Corporation/Microstft Corp.
XXXXXX Bytes free
Ok
```



備　考　固定ディスクが接続されている場合でも，ユーザ識別名を使用しないモードで立上げることができます．

メモリスイッチSW5 2^2ビットをONにしておきます．

この場合，User identifier?のメッセージは表示されません．

ユーザ識別名を使用しない場合，固定ディスク内のファイルはすべて「共通ユーザ識別名」で管理されます．したがって，すべてのファイルはどのユーザからも参照/更新することができます．

メモリスイッチの詳細については「第21章　メモリスイッチ」を参照してください．

1.2.2　固定ディスクからの立上げ

固定ディスクからDISKモードBASICを立上げるためには，システムディスクからDISK CODE部とIPLを，あらかじめ固定ディスクに移しておく必要があります．これは“sysgen.nip”ユーティリティを用いて行います．“sysgen.nip”ユーティリティの使用方法については，「第15章　システムディスクに格納されているユーティリティプログラム」を参照してください．

次の場合，固定ディスクからDISKモードBASICが立上がります．

(1) 固定ディスクにシステムディスクのDISK CODEとIPLを登録しておきます．

(2) メモリスイッチSW5の2^4, 2^5, 2^6, 2^7 ビットがすべてOFF（ゼロ）の場合（標準の状態），システムはフロッピィディスク装置→固定ディスク装置の順で適正なIPLを探策していきます．したがって，いずれのフロッピィディスク装置にもシステムディスクがセットされていない場合，固定ディスクからDISKモードBASICが起動されます．

(3) メモリスイッチSW5のビット2^5, 2^7ビットがON(1)で，2^4, 2^6がOFF（ゼロ）の場合，固定

ディスクのみを立上げ装置とみなします．フロッピィディスク装置にフロッピィディスクがセットされていてもいなくても，この場合，固定ディスクからDISKモードBASICが起動されます．

固定ディスクからシステム立ち上げを行った場合のユーザ識別名の扱いおよび表示メッセージ等は，システムディスクから立ち上げを行った場合と同じです．

1.2.3 オートスタート

今までに説明した通常のシステム立ち上げ方法ですと，システムの立ち上げ時に，How many files情報やユーザ識別名，さらに実行すべきコマンドやステートメント（RUNコマンド）を入力する必要があります．

これらの情報を，あらかじめシステムに覚えさせておくことにより，立ち上げ時のキー入力を不要にし，システム立ち上げからプログラム実行までを簡便に行う方法があります．

これを，**オートスタート**と呼びます．

システムディスクから立ち上げる場合と，固定ディスクから立ち上げる場合があります．それぞれについて説明します．

(1) **システムディスクから立ち上げる場合**

オートスタートを行うためには，あらかじめ，システムディスクの特定の場所（IDと呼ぶところ…「12.6　ID」参照）にHow many files情報や実行すべきBASICコマンドまたはステートメントを書き込んでおかなければなりません．

これは，“setinf.n88”というユーティリティプログラムを用いて行います．詳細は「第15章　システムディスクに格納されているユーティリティプログラム」を参照してください．

固定ディスクを使用する場合，システム立ち上げ時に，ユーザ識別名の問い合わせが行われますが，この情報はID部へ登録できません．したがって，このままで，システムを立ち上げますと，How many files(0－15)?に対する入力は不要ですが，User identifier?の問合わせには答えなければなりません．

これも不要にするには次の２つの方法があります．

(1) メモリスイッチSW5 2^2ビットを「ON」にしておく

こうしておくと，ユーザ識別名の問い合わせは行われず，ユーザ識別名として“999”を入力したものとして扱われます．

(2) “999”以外のユーザ識別名を与える必要がある場合は，次の処置を行って下さい．

ID部を設定するオートスタート時の実行ステートメントの先頭に次に示す命令を設定しておいて下さい．

DEF SEG=&H60 : POKE &H510, n_1 : POKE &H511, n_2 : POKE &H512, n_3

ここで，n_1,n_2,n_3は次に説明する数値です．

n_1：ユーザ識別名の先頭１文字の整数表記

n_2：ユーザ識別名の２文字目の整数表記

n_3：ユーザ識別名の３文字目の整数表記

例(1)　ユーザ識別名が“A01”の場合

DFF SEG=&H60 : POKE &H510, 65 : POKE &H511, 48 : POKE &H512, 49

例(2)　ユーザ識別名が“a△△”の場合（△は空白を意味する）

DEF SEG=&H60 : POKE &H510, 97 : POKE &H511, 32 : POKE &H512, 32

注　意　"setinf. n88" ユーティリティを実行する前に，"sysgen. nip" ユーティリティを使って，固定ディスク上に，システムディスクのDISK CODE部とIPLを移しておく必要があります．

その他のことがらについては，システムディスクから立ち上げる場合と同じです．

1.2.4　システムのリセット

本体の前面にリセットスイッチがついています．このスイッチを押すと，電源を「ON」にしたときと同じ状態にセットされます（ハードウエアの諸機能が初期化されます）．このリセットスイッチは次のような場合に限り使用します．

1．　キーボードから何も入力できなくなったとき
2．　システムの新たな起動をおこないたい時（たとえば，BASICを使用している状態から，MS-DOSシステムに切り換えたい時がそうです）．

リセットスイッチを押すと，利用者メモリ領域はすべてクリアされますから，入力されていたプログラムやデータは消えてしまいます．リセットスイッチの使用には十分注意してください．

1.2.5　ウォームリスタート（warm restart）

BASICを使用している場合に，STOPキーを押しながら，リセットスイッチを押すと（リセットスイッチを離してから，STOPキーを離してください），画面がクリアされ，左上スミに「OK」と表示されます．この機能を「ウォームリスタート」といいます．ウォームリスタートは，リセットスイッチを押したときと，ほぼ同じ機能を果しますが，入力されたプログラムやデータは保存されます．

何らかの理由でシステムがハングアップした場合（通常のキー入力ができない状態），ウォームリスタートをおこなうと，ハングアップ前の状態にもどすことができます．

この場合，直ちにプログラムをディスクに保存し，システムをリセットしてください．

注　意　ウォームリスタートを行うと，テキスト画面モードは常に80桁×25行モードになります．以前のシステム状態には関係しません．

第2章
DISKモードBASIC使用時の種々の機能選択

DISKモードBASIC使用時，次に示す種々の機能選択が可能です．

(1) グラフィック機能の選択（基本グラフィックモードと拡張グラフィックモードのいずれかの選択）

(2) 日本語入力方法の選択（連文節変換入力方式，単文節変換入力方式あるいはJISコード入力方式のいずれかの選択）

(3) 画面ハードコピー機能の選択（従来のPC-9800シリーズ各種に備わっていた画面ハードコピー機能と同等のものと，拡張画面ハードコピー機能のいずれかの選択）

(4) モニタモードにおけるディスクモード関連機能（たとえば，アセンブル/ディスアセンブル機能，ダンプディスク，エディットメモリ等の機能）の使用の有無

(5) モデム-NCU内蔵電話機制御機能の使用の有無

これらの機能選択は本体前面のディップスイッチあるいはメモリスイッチ等で指定します．

また，どの機能が選ばれるかはシステムの起動時に決まります．

ディップスイッチに関しては「ハードウェアマニュアル」を，また，メモリスイッチに関しては「第21章　メモリスイッチ」を参照して下さい．

2.1 基本グラフィックモードと拡張グラフィックモードの選択

本体前面のディップスイッチSW1の8番スイッチ（SW1の右端のスイッチです．このスイッチを拡張グラフィックスイッチと呼びます）がOFFの場合，基本グラフィックモードが選択されます．

ONの場合（下に倒れた状態がONです），拡張グラフィックモードが選択されます．

このスイッチは本体出荷時ONになっています．

(1) 基本グラフィックモードと拡張グラフィックモードの機能差

	基本グラフィックモード	拡張グラフィックモード
グラフィック機能の差	使用できるカラーの種類はシステムが規定した8色です	使用できるカラーの種類はシステムが規定した8色に加え、4096色中の任意の8色あるいは16色の中間色表示が可能になります
グラフィック描画のスピード	PC-9801E/F/M並みの描画スピードです	高速描画のためのハードウェアを使用し動作しますので，高速描画が可能になります

利用者メモリの圧迫	利用者メモリは特に圧迫しません	22KBのシステムコードが利用者メモリ上にロードされますので、その分利用者メモリが減ります

注意 (1) 4096色中の任意の8色あるいは16色の表示は，接続されているディスプレイ装置がアナログRGB対応ディスプレイの場合のみ可能です．

(2) 利用者メモリとは，プログラム実行時に使用される変数の値や配列の値を格納するメモリ領域のことです．詳細については「22.2 N88-BASIC(86)のメモリ構成」を参照してください．

(2) 基本グラフィックモードと拡張グラフィックモードの選択基準

・アナログRGB対応ディスプレイを接続している場合，4096色中の8色あるいは16色の中間色表示が可能です．この場合は拡張グラフィックモードを選んで下さい．

・メモリに余裕がない（拡張グラフィックモードを選択すると利用者メモリが22KB減ります）あるいはグラフィックの描画スピードが速すぎて困る場合等には基本グラフィックモードを選んで下さい．

基本グラフィックモードと拡張グラフィックモードの詳細については「7.2.2　基本グラフィックモードと拡張グラフィックモード」および「8.2　カラー指定」を参照して下さい．

備考 ROMモードBASIC使用時は無条件に基本グラフィックモードとなります．

2.2　日本語入力方法の選択

次に示す3種類の中から使用する日本語入力方法を選ぶことができます．

1．連文節変換入力方式による入力
2．単文節変換入力方式による入力
3．JISコード入力方式による入力

各方式の特徴は次の通りです．

特長＼入力方式	概　　要	日本語入力の簡便さ	メモリ/ファイル所要量
連文節変換入力方式	“読み”→“日本語”の変換を複数の文節に渡っておこないます．ただし、利用者メモリの圧迫が単文節変換入力方式よりも大きくなります（単文節変換入力方式の66KB注に比べ114KB注のメモリが必要です）．	簡　単 ↑	大きい ↑
単文節変換入力方式	“読み”→“日本語”の変換を文節単位におこないます．連文節変換入力方式に比べ利用者メモリの圧迫が小さくてすみます．（66KB注必要です）．	↓	↓
JISコード入力方式	単字単位に対応する日本語のJISコードを入力していきます．利用者メモリを圧迫しません。辞書ファイルも使用しません．	面　倒	小さい

注：基本システムコード24KBを含んでいます．

日本語入力の方法はシステムディスクで一意に決ります．

システムディスクは連文節変換入力用に特殊化された状態で出荷されています．

したがって，標準の状態で単文節変換入力あるいはJISコード入力方式による日本語入力をおこなうことはできません．

単文節変換入力あるいはJISコード入力をおこないたい場合，“setup.n88”ユーティリティを使用し，システムディスクの属性変更をおこないます．

つまり，単文節変換入力用のシステムディスクあるいはJISコード入力用のシステムディスクをそれぞれ別に作成するわけです．

単文節変換入力用に特殊化したシステムディスクを使用し，システムを立上げた場合，連文節変換入力あるいはJISコード入力をおこなうことはできません．

同様に，JISコード入力用に特殊化したシステムディスクを使用し，システムを立上げた場合，連文節変換入力あるいは単文節変換入力をおこなうことはできません．

“setup.n88”ユーティリティの詳細については，「第15章　システムディスクに格納されているユーティリティプログラム」を参照してください．

備　考　(1)　連文節変換入力方式，単文節変換入力方式のいずれの場合も同一の辞書ファイルを使用します．辞書ファイルの標準の名前は“BUNSETSU”です．

(2) ROMモードBASIC使用時にはJISコード入力方式の日本語入力しか行えません.
日本語入力方法の詳細については「第5章　日本語入力」を参照して下さい.

2.3 画面ハードコピー機能の選択

画面ハードコピー機能はディスプレイ画面に表示された画面のイメージをそのままプリンタに出力する機能です.
通常の画面ハードコピー機能は次のようになっています.

項番	キー操作	COPY文	機　　能
1	CTRL + COPY	COPY 1	テキスト画面のみの画面ハードコピーを行います。出力される文字は接続されているプリンタの印字体となります。たとえば明朝体印字が可能なPC-PR201系プリンタを使用しますと明朝体による印字となります.
2	GRPH + COPY	COPY 2	グラフィック画面のみの画面ハードコピーを行います。
3	COPY	COPY 3	テキスト画面とグラフィック画面を重ねてプリンタに出力します。(項番1と項番2の出力情報をそのまま重ねたようなイメージです)
4	——	COPY 4	グラフィック画面の縮小コピー(縦方向に縮小します)を出力します。グラフィック画面が640×200ドットモードの場合、縦方向に縮小された形で出力されますので、グラフィック画面に表示した漢字等が見易くなります。
5	——	COPY 5	項番3の両画面コピーをそのまま縦方向に縮小したコピーです。項番4と同様、640×200ドットモード時に有効です。

メモリスイッチSW6・2^4ビットをONにした状態でシステムを立上げますと拡張画面ハードコピー機能が使用できます(メモリスイッチの設定には，システムディスクに格納されているユーティリティ「switch.n88」を御使用ください).
拡張画面ハードコピー機能を選択しますと，次のような画面ハードコピー機能が使用できます.

(1)　グラフィック画面とテキスト画面の完全合成画面ハードコピーがとれます．

項番	キー操作	COPY文	機　　能
1	CTRL ＋ COPY	COPY 1	テキスト画面のみの画面ハードコピーを行います
2	GRPH ＋ COPY	COPY 2	グラフィック画面ハードコピーを行います
3	COPY	COPY 3	両画面を完全合成したイメージの画面ハードコピーを行います（通常の画面ハードコピーの場合，テキスト画面の情報はコード出力（プリンタの印字体で出力）となりますが，拡張画面ハードコピーの場合，テキスト画面情報はグラフィック画面情報と合成された後，ドット情報として出力されます）
4	──	COPY 4	──
5	──	COPY 5	──

(2)　PC-PR201V等のカラープリンタが接続されている場合，カラー画面ハードコピーを出力することができます．

項番	キー操作	COPY文	機　　能
1	CTRL ＋ COPY	COPY 1	テキスト画面のみのカラー画面ハードコピーを行います
2	GRPH ＋ COPY	COPY 2	グラフィック画面のみのカラー画面ハードコピー（白黒反転）を行います
3	COPY	COPY 3	両画面を合成したイメージのカラー画面ハードコピー（白黒反転）を行います
4	──	COPY 4	グラフィック画面のみのカラー画面ハードコピー（白黒反転せず）を行います
5	──	COPY 5	両画面を合成したイメージのカラー画面ハードコピー（白黒反転せず）を行います

画面ハードコピー機能の詳細に関しては「第9章　画面ハードコピー」を参照して下さい．

備　考　ROMモードBASIC使用時においては拡張画面ハードコピー機能は使用できません．

注　意　拡張画面ハードコピーを選択しますと，6KBのシステムコードが利用者メモリ上に追加ロードされます．その分，利用者メモリが減りますので御注意下さい．

2.4　モニタモードにおける拡張機能の選択

モニタモードは機械語プログラムの作成/デバッグのための環境を利用者に提供します．
モニタモードに入るためのコマンドとしてMONコマンドが用意されています．

モニタモードで使用可能な機能は次の通りです.

項番	コマンド	意　　味	機　　　能
1	**A**	**アセンブル**	**入力した1行をアセンブルします**
2	B	ベース	数値の表現形式を変えます（8進→16進）
3	C	チェンジセグメントベース	機械語プログラムセグメントのベースアドレスを設定します。
4	D	ダンプ・メモリ	メモリの内容をディスプレイに表示します
5	**E**	**エディト・メモリ**	**スクリーンエディタの機能を用いてメモリの内容を変更します**
6	F	フィル・メモリ	メモリの内容を定数で埋めていきます
7	G	ゴー	ユーザープログラムを実行します
8	I	インプット	I/Oポートの値を読み込みます
9	**L**	**ディス・アセンブル**	**機械語を逆アセンブルします**
10	M	ムーブ・メモリ	ある範囲のメモリの内容を他のアドレスのメモリ領域へ移します
11	O	アウトプット	I/Oポートへデータを出力します
12	P	プリンタ・スイッチ	プリンタへの出力をコントロールします
13	R	リード・テープ	カセットテープからデータをロードします
14	S	セット・メモリ	メモリにデータをセットします
15	TM	テスト・メモリ	メモリをテストします
16	V	ベリファイ・テープ	カセットテープの内容と、メモリの内容を比較します
17	W	ライト・テープ	メモリの内容をカセットテープにセーブします
18	X	イグザミン・レジスタ	CPUのレジスタの値を調べ、変更します
19	**HELP または CTRL +A**	**ヘルプ**	**コマンドとそのパラメータの形式をディスプレイに表示します**
20	CTRL +B	リターン	BASICモードへ復帰します
21	**CTRL +D**	**ダンプ・ディスク**	**ディスクの内容をディスプレイに表示します**
22	**CTRL +R**	**リード・ディスク**	**ディスクからデータをロードします**
23	**CTRL +W**	**ライト・ディスク**	**ディスクへデータをセーブします**

これらの機能のうち[　]で示されている機能はDISKモードBASICの環境でだけ使用できるものです.

通常，これらの機能は，プログラムの実行のために必要としませんので，標準では備わっていません．使用したい場合にはメモリスイッチSW6・2^3ビットをONにしてからシステムを起動します．

注　意　モニタモードの拡張機能の使用をメモリスイッチで宣言しますと，11KBのシステムコードが利用者メモリ上に追加ロードされます．その分利用者メモリが減りますので御注意下さい．

モニタモードに関する詳細については「第14章　モニタモード」を参照して下さい．

2.5　モデム-NCU内蔵電話機制御機能の選択

N_{88}-日本語BASIC(86)は，モデム-NCU内蔵電話機PC-TL101，PC-TL102オートホン，モデムボード＋ハンドセットあるいはインテリジェントモデムDATAX ITM1200，DATAX ITM1212等の制御機能を持っており，オートダイアル，自動着信さらにはモデムを利用したデータ通信等を簡単に実現することができます．メモリスイッチSW6・2^5ビットをONにした状態でシステムを立上げますと，次に示す拡張電話制御命令が使用可能になります（メモリスイッチの設定には，システムディスクに格納されているユーティリティ「switch.n88」を御使用ください）．

項番	コマンド名あるいは関数名	機　　能	コマンド/関数
1	CMD BREAK	ブレーク信号を送出する	コマンド
2	CMD CHANGE DUPLEX	通信方式を切り換える	コマンド
3	CMD DIAL	電話をかける	コマンド
4	CMD ERROR ON/OFF/STOP	通信エラー割り込みを制御する	コマンド
5	CMD LINE CLOSE	電話機とBASICを論理的に切り離す	コマンド
6	CMD LINE ON/OFF/STOP	着信割り込みを制御する	コマンド
7	CMD LINE OPEN	電話機とBASICを論理的に接続する	コマンド
8	CMD MODE CUT	電話を切る	コマンド
9	CMD ON ERROR GOSUB	通信エラー時の割り込み先を定義する	コマンド
10	CMD ON LINE GOSUB	着信があった時の割り込み先を定義する	コマンド
11	CMD RECEIVE	着信があった場合の動作を指定する	コマンド
12	CMD STORE DIAL	電話機に電話番号を記憶させる	コマンド
13	STATUS DIAL	電話機に記憶されている電話番号の機能を調べる	関数
14	STATUS DIAL$	電話機に記憶されている電話番号を調べる	関数
15	STATUS ERROR	通信エラーの有無/種類を調べる	関数
16	STATUS LINE	着信の有無を調べる	関数
17	STATUS MODE	現在の電話機のモードを調べる	関数

モデム-NCU内蔵電話機制御機能の詳細については「19.1　モデム-NCU内蔵電話機制御機能」を参照して下さい．

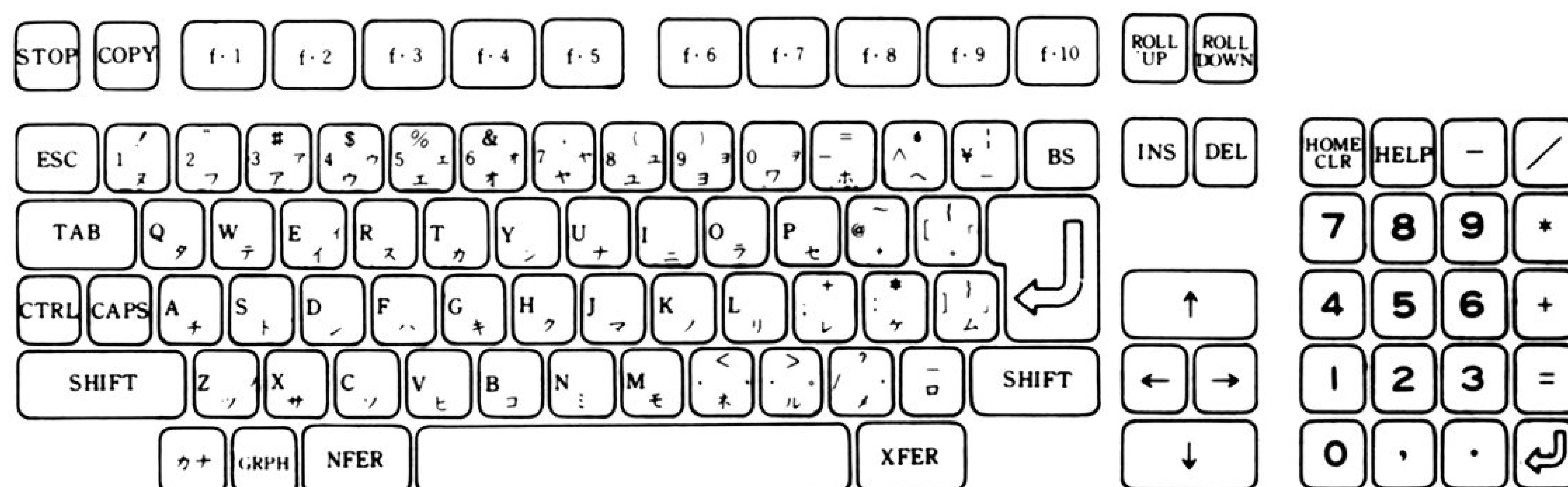
注　意　拡張電話制御命令の使用を宣言しシステムを起動しますと，システムコードが20KB増大します．その分利用者メモリが減少しますので御注意下さい．

第3章

キーボード

3.1 キーの配置

キーボードは，JIS配列に準拠しています．リターンキー（↵）やシフトキー（SHIFT）など，良く使われるキーは押しやすい位置に，他のキーより大きなキーで配置されています．また，N88-BASICを使用しているときは，ヘルプキー（HELP）やコピーキー（COPY）など特別な機能を持ったキーが使用できます．さらに，リピート機能が備わっていますから，同じキーを約0.5秒間押し続けると，同じ文字が連続して表示されます．キーボードの右側には数値入力用キー（テンキー）が備わっています．多くの数値を入力する場合に非常に便利です．



キー配列

3.2 キー入力

キー入力には，ノーマルモード，グラフィックシンボル入力モード，カナ入力モード，日本語入力モードの4つのモードがあります．

(1) ノーマルモード

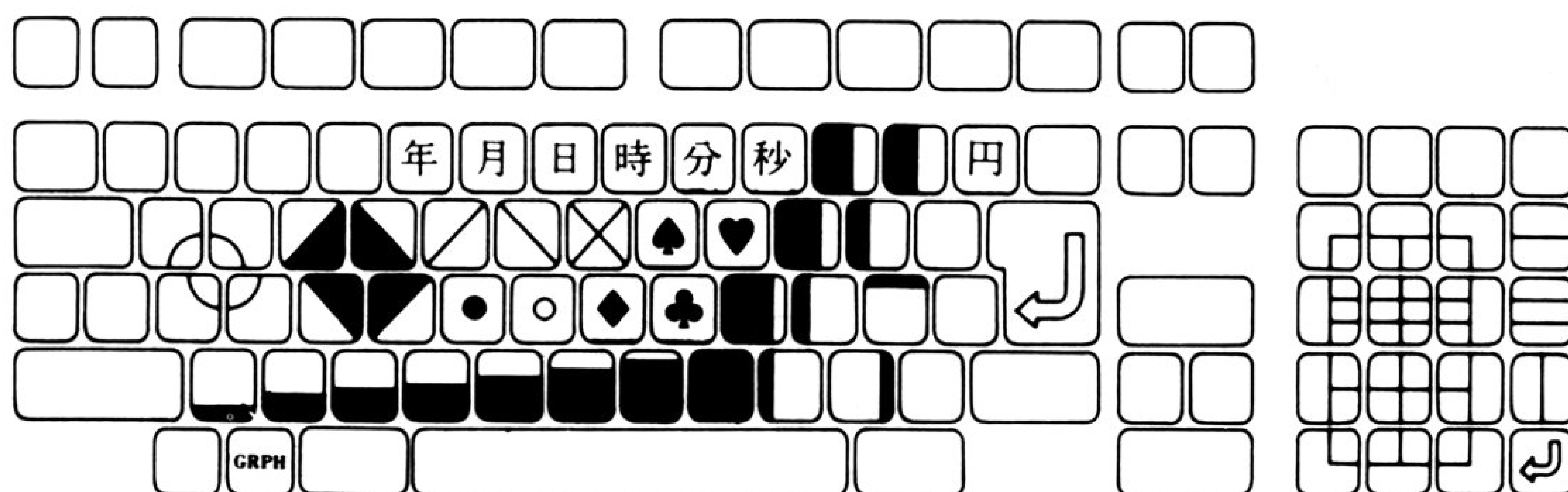
キーボードの通常の状態での入力で，シフトキー（SHIFT）を押したシフトポジションと合わせて，アルファベットや数字，特殊記号（+，＊，?.）など，ASCII標準キャラクタを入力するモードです．このとき，カナキーはロックされていない（押されたままの状態になってない）ことに注意してください．

アルファベットは，そのまま押すと小文字が入力され，シフトキーを押しながら入力すると大文字が入力されます．CAPSキーを1度押すと，キーはロックされ，シフトキーを押さない状態でアルファベットの大文字が入力され，シフトキーを押しながら入力すると小文字が入力されま

す．この CAPS キーは，BASICのプログラムなどで，大文字を続けて入力する場合に使用すると大変便利です． CAPS キーのロックを解除する場合は，もう一度， CAPS キーを押します．

(2) グラフィックシンボル入力モード

GRPH キーを押しながら，キー入力を行うモードです．例えば， GRPH キーを押しながら， O ラ キーを押すと，“♥”のマークが入力されます．入力できるグラフィックシンボルは，次の「グラフィックシンボルのキー配列」を参照してください．グラフィックシンボル入力モードは， カナ キーや CAPS キーがロックされている状態でも動作します．



グラフィックシンボルのキー配列

(3) カナ入力モード

カナ キーを押してロックした状態で，キー入力を行います．また，ツ，ア，ウ，エ，オなどの右肩にあるカナ文字やカナ記号を入力した場合は， SHIFT キーを押しながら入力します．（詳しくは，「3.3 特殊キーの説明」を参照してください．）

(4) 日本語入力モード

変換キー XFER キーと CTRL キーとを同時にキー入力すると日本語入力モードになります．その時のシステムの状態により，文節変換入力，表示選択入力，JISコード入力のいずれかの入力方法で日本語の入力ができます．

3.3 特殊キーの説明

SHIFT （シフトキー）

タイプライタのシフトキーと同じです．各キーのシフトポジションにある文字を入力する場合に使用します．また，アルファベットを入力する場合は，大文字の入力に使用し， CAPS キーがロックされている状態では，逆に小文字の入力に使用します．

CTRL （コントロールキー）

他のキーと組み合わせて，特殊なコードを入力します．

キー入力	動　　　作
CTRL ＋ A チ	HELP と同じ機能を果します．
CTRL ＋ B コ	カーソルを１項目ごと左へ移動させます．
CTRL ＋ C ソ	プログラムの実行を中断してBASICのコマンドに戻ります

CTRL + Dシ	現在，カーソルが点滅している項目を消去します．
CTRL + Eイ	現在，カーソルが点滅している位置からその行の終りまでを消去します．
CTRL + Fハ	カーソルを次の項目の先頭に移します．
CTRL + Gキ	ブザー（ビープ音）を鳴らします．
CTRL + Hク	カーソルの前の1文字を消去します．DEL と同じ機能です．
CTRL + Iニ	タブ位置までカーソルを移動させます．タブ位置は8文字ごとに設定されています．
CTRL + Jマ	ラインフィード（LFコード）を挿入します．インサートモードでは，行の分割機能があります．（「第4章　スクリーンエディタ」参照）．
CTRL + Kノ	カーソルをホームポジション（左上スミ）に移動させます．SHIFT + HOME CLR と同じです．
CTRL + Lリ	テキスト画面をクリアします．HOME CLR と同じです．
CTRL + Mモ	リターンキー（↵）と同じです．
CTRL + Oラ	テキスト画面の表示を中止します．再び CTRL + Oラ を入力すると再開されます（中止してから再開されるまでに表示されたデータは失われます）．
CTRL + Rス	インサートモードになります．INS と同じです．
CTRL + Sト	プログラムの実行の一時停止，およびlist文，llist 文の実行の一時停止．STOP，COPY，SHIFT，カナ，GRPH，CTRL + Cソ 以外のキーを入力すれば実行が再開されます．
CTRL + Uナ	カーソルが点滅している行を一行全部消去します．
CTRL + Xサ	カーソルが点滅している行の最後の文字の次にカーソルを移動させます．

↵ リターンキー（RETURN）

↵ このキーの入力により，1行の入力が終ります．このキーを入力するとカーソルは，次の行の先頭に移動します．

カナ カナキー

このキーを押すと，キーはロックされ，入力はカナモードになります．キーに表示されている文字の位置により SHIFT キーと カナ キーの組み合わせを表わしています．

ロックされたカナキーをもう一度押すと，[カナ] キーのロックは解除されます

例　[3#ア]→3

[SHIFT] ＋[3#ア]→#

[カナ] ＋[3#ア]→ア

[カナ] ＋[SHIFT] ＋[3#ア]→ァ

[Aチ]→a

[CAPS] ＋[Aチ] →A

[SHIFT] ＋[Aチ]→A

[CAPS] ＋[SHIFT] ＋[Aチ]→a

[カナ] ＋[Aチ]→チ

[STOP]　**ストップキー**

プログラムの実行を停止するときに使用します．

ターミナルモード時にはブレークキー（ブレーク信号を送出する）として使用します．

[HOME CLR]　**ホーム・クリアキー**

このキーを入力するとテキスト画面のスクロール画面（スクロール画面については第4章スクリーンエディタを参照してください）がクリアされ，カーソルはスクロール画面の先頭に移ります．[SHIFT] キーを押しながら入力すると，カーソルだけが移動します．

[→] [←] [↑] [↓]　**カーソル移動キー**

カーソルを移動するときに使用します．

画面の右端にカーソルがあるとき，[→] キーを入力するとカーソルは次の行の左端に移動します．画面の右下スミにあるときに，[→] キーを入力すると画面は1行上へスクロールし，カーソルは最下位の行の左端に移動します．画面の左端にカーソルがあるとき，[←] キーを入力するとカーソルは1つ前の行の右端に移ります．最上位行の左端にあるときは，移動しません．画面の最上位行または，スクロール画面の先頭行にカーソルがあるとき，[↑] キーを入力してもカーソルは移動しません．画面の最下位行にカーソルがあるとき，[↓] キーを入力した場合も同じです．

[INS]　**インサートキー**

このキーを入力すると，インサートモードになり，カーソルが，点滅している文字とその前の文字との間に，入力された文字を挿入します．もう一度，[INS] キーを入力するか，カーソル移動キー，もしくはリターンキーを入力するとインサートモードから抜け出します．

[DEL]　**デリートキー**

このキーを入力すると，点滅しているカーソルの左の文字が1文字消され，カーソルを含んだカーソルより右の1行分の文字列を左につめていきます．

[CAPS]　**キャピタルロックキー**

このキーを押してロックすると，それ以後，アルファベットを入力すると大文字が入力されます．このとき [SHIFT] キーを押しながら入力すると小文字が入力されます．[CAPS] キーのロックを解除する場合は，もう一度このキーを押します．

[f･1] ～ [f･10]　**ファンクションキー**

10種のファンクションキーが使用できます．

各ファンクションキーには標準値として次に示す文字列がそれぞれ割当てられています．

f･1	load"	f･6	save"
f･2	auto	f･7	key
f･3	go to	f･8	print
f･4	list	f･9	edit.$^C{}_R$
f･5	run$^C{}_R$	f･10	cont$^C{}_R$

このファンクションキーを押すと，定義されている文字列が入力されます．ですから，よく使う文字列をセットしておけば，プログラムやデータなどを入力する時間がずっと短くなります．各ファンクションキーは，最大15文字まで定義できます．定義されている内容の表示は，40キャラクタモードのとき5個でそれぞれは6文字，シフトキーを押さないとf･1〜f･5，シフトキーを押すとf･6〜f･10が表示されます．80キャラクタモードのとき10個で，それぞれ6文字表示されます．

ファンクションキーを新しく定義する場合は，次のように入力します．

key 4, "run" + chr$(13) [↵]

これで[f･4]に，"run$^C{}_R$"が定義されます．(**key**文および**chr $**については，「BASICリファレンスマニュアル」を参照してください.)

また，現在，定義されている各ファンクションキーの内容を表示させるには，次の命令を実行します．

key list [↵]

[GRPH]　**グラフキー**

グラフィックシンボルを入力するときに使います．

[ESC]　**エスケープキー**

エスケープコード（1B)$_{16}$を入力するときに使用します．

[ROLL UP]　**ロールアップキー**

[ROLL DOWN]　**ロールダウンキー**

EDIT文とともに用いて，スクリーンエディタによるプログラムの修正に使用します（DISK BASICモードの時に有効な機能です)．詳しくは，「第4章　スクリーンエディタ」を参照してください．

また，機械語モニタのEコマンドでも，同様の用途に使用します．ROM BASICモードの場合，[ROLL UP] [ROLL DOWN] キーによりテキスト画面を上下にスクロールさせるだけです．

[TAB] **タブキー**（水平タブキー）

このキーを入力すると，カーソルは8桁単位で右へ移動します．

[COPY]　**コピーキー**

画面のハードコピーをプリンタに出力することができます．詳しくは，「第9章　画面ハードコピー」を参照して下さい．

[HELP]　**ヘルプキー**

ヘルプキーには，2つの機能があります．

1．エラー位置の検出

プログラムの実行中にエラーが起きた場合に，その文と，場所を表示します．(詳しくは，「第4章　スクリーンエディタ」を参照してください.)

2．ON HELP GOSUB文と組合せて使用します．

(詳しくは，「BASICリファレンスマニュアル」を参照してください.)

BS バックスペースキー

バックスペースキーには，２つの機能があります．

1．ターミナルモードのとき

半２重モードでは BS キー$(08)_{16}$コードを送信し，同時にカーソルの前（左）の１文字が消去されます．

2．ターミナルモード以外のとき

DEL キーと同じ動作をします．

XFER 変換キー

日本語入力制御キーとして使用されます．使用法については「第５章　日本語入力」を参照して下さい．

NFER 無変換キー

日本語入力制御キーとして使用されます．使用方法については「第５章　日本語入力」を参照して下さい．

第4章
スクリーンエディタ

N88-BASICは，プログラムの作成および修正が簡単に行えるように，「スクリーンエディタ」と呼ばれる画面編集機能を備えています．この機能により，プログラムの入力や訂正が短時間で行えます．

4.1 入力中の修正

キーボードからプログラムなどを入力しているとき，誤って隣のキーを押してしまったり，途中で文字を抜かしてしまったりする場合があります．このような時，その行を最初から入力しなおすのは，大変わずらわしいことです．そこで，スクリーンエディタの機能を使います．

(1) 直前の文字の修正

DEL を押します．すると，カーソルの前の1文字が消去されます．CTRL ＋ Hク でも同じ機能を実現できます．

(2) 途中の文字の修正

途中で押しまちがえた文字を正しい文字に置き換えたい場合は，カーソル移動キー（← → ↑ ↓）を使用して，まちがった文字のところへ移動させ，正しい文字に置き換えた後，再びカーソル移動キーで入力を中断したところへ戻って入力を続けます．

(3) 途中の文字の消去

消去したい文字の次の文字に，カーソルを移動させた後，DEL を入力します．この場合，カーソルから右の文字列が1文字，左へ移動します．

(4) 文字の挿入

何文字か途中で挿入したい位置の次の文字にカーソルを移動させ，INS を入力します．これでインサートモードとなり，それ以後に入力される文字をカーソルの前に挿入していきます．インサートモードから抜け出すには，再び，INS を入力するか，カーソル移動キーかリターンキー（↵）を入力します

（INS は，CTRL ＋ Rス で代用できます）

スクリーンエディタの機能を用いてプログラムの訂正を行った場合は，各行番号の文の訂正が終る度に，リターンキー（↵）を入力してください．BASICは，リターンキーの入力により，その文の変更を理解します．

NOTE 1つの行番号の文での，入力や訂正が終ったら，他の行番号の文に移る前に，必ずリターンキー（↵）を入力してください．

リターンキーを入力する時，カーソルをいちいち文の終端まで移動する必要はありません．カーソルが文のどの位置にあっても，文全体が変更の対象になります．

(例)「10 prannt Hello」を「10 print "Hello"」に訂正してみましょう．(□はカーソルを表わし

ています)

10 prannt Hello□

[←]でカーソルを移動させます.

10 pr[a]nnt Hello

[I] を入力します.

10 pri[n]nt Hello

[→]でカーソルを移動させます.

10 prin[n]t Hello

[DEL] で,カーソルの前の “n” を消去します.

10 pri[n]t Hello

[→]でカーソルを移動させます.

10 print [H]ello

[INS] で,インサートモードにした後,

[SHIFT]+[2"]を入力します.

10 print “[H]ello

[→]でカーソルを移動させます.これで,インサートモードから抜け出しています.

10 print “Hello □

[SHIFT] + [2"] を入力します.これでOKですね.

10 print “Hello” □

他の行番号の文へ移る前に,リターンキー[↵]を入力します.これで,訂正した後の文が,メモリに記憶されました.

4.2 エラーの訂正

プログラムに何らかのエラーがある場合,その文で実行が中断されます.このとき,エラーの種類と,そのエラーが起きた行番号がディスプレイに表示されます.さらに,エラーの種類によっては,その行が表示され,エラーのある項目か,その次の項目の先頭で,カーソルが点滅します.

20 PRINT A :PLINT [B]:GOTO 10

この場合は,カーソルのすぐ前の項目,すなわち “PLINT” の中にエラーがあります.“R” が,“L” とまちがって入力されています.そこで,[←]でカーソルを移動させ,[R]を入力して,リターンキーを押して訂正を終ります.

エラーのあった行を表示したい場合は,[HELP] を入力します.すると,エラーメッセージで表示された行番号の文が表示され,エラーのある項目か,その次の項目でカーソルが点滅します.

エラーの原因が,その行ではなく,他の行にある場合があります.そのような場合は,edit 文を使います.

edit 〈エラーのあった行番号〉 [↵],又は [f·9]

指定した行が画面の最上位に表示され,エディットモードに入ります.エディットモードでは[ROLL UP],[ROLL DOWN] を入力することにより,巻き物を見るように,画面に表示されている文の前後が表示されます.この機能を利用して,訂正する文を画面に表示させ,訂正します.この場合も,訂正が

終ったら，他の行番号の文に移る前に，リターンキーを必ず入力してください．エディットモードから抜け出すには，[HOME CLR]を入力します．

4.3 便利な編集機能

(1)　1行消去　[CTRL]＋[Uナ]

カーソルが点滅している行を消去する場合は，[CTRL]＋[Uナ]を入力します．その行の先頭から，行の最後までを画面から消去します．この場合は，行番号も消えてしまいますからリターンキーを入力しても，プログラムは訂正されません．

(2)　カーソルから，その行の終りまでの消去　[CTRL]＋[Eィイ]

カーソルが点滅している文字から，その行の終りまで消去する場合は，[CTRL]＋[Eィイ]を入力します．この後で，リターンキーを入力すれば，メモリの内容も訂正されます．ですから行番号のすぐ次の位置にカーソルを移動し，[CTRL]＋[Eィイ]を入力し，その後リターンキーを入力すると，その行番号の文はプログラムから削除されます．

(3)　カーソルの移動　[CTRL]＋[F、]，[CTRL]＋[Bコ]，[CTRL]＋[Xサ]

キーボードには，オートリピート機能が備わっているため，カーソル移動キーを押し続けると，カーソルは次々と移動します．しかし，さらに便利な機能として，項目ごとの移動ができます．

[CTRL]＋[F、]を入力すると，カーソルは次の項の先頭へ移動します．また，[CTRL]＋[Bコ]を入力すると1つ前の項目の先頭へ移動します．また，[CTRL]＋[Xサ]を入力すると，その行の最後の文字の次にカーソルが移動します．

(4)　行の分割とラインフィードコードの挿入　[CTRL]＋[Jマ]

1行の内容を2行にわたるように書き直したい場合には，区切りたい箇所でカーソルを止め，[INS]でインサートモードに切り換え，[CTRL]＋[Jマ]を押すと，カーソルから後のその行の最後までの文が，次へ移動します．移動した行の先頭に行番号を付加し，[↵]を押せば，これまでの1行の内容が2行にわたって書かれます．

また，インサートモードになっていないときに，[CTRL]＋[Jマ]を入力すると，その行の最後にラインフィードコード（LF）を挿入します．

(5)　1項目の削除　[CTRL]＋[Dシ]

カーソルの点滅している文字から，その項目の終りまでを消去します．

注　意　エディットモードで[ROLL UP]，[ROLL DOWN]を使ってプログラムをスクロールしている状態で，追加行を入力するためには，1つの方法として，まずスクロール画面上の適当な行番号をもった行を[CTRL]＋[Eィイ]で画面上から最初の桁から最後の桁まで消去した上で，あらためて消去したあきエリアを使って，行番号付文をキーインし，リターンキー[↵]を入力することによって可能です．

第5章
日本語入力

スクリーンエディタを使用してプログラムを編集しているときやINPUT文の実行によりキー入力が可能な状態にある場合，CTRL＋XFER（CTRLキーを押しながらXFERキーを押します）を入力することにより日本語の入力が可能になります．日本語入力が可能な状態を日本語入力モードと呼びます．

また，KINPUT文を実行すると，自動的に日本語入力モードとなります．

日本語入力の方法には次の3つがあります．

1　連文節変換入力方式による入力
2　単文節変換入力方式による入力
3　JISコード入力方式による入力

各方式の特徴は次のとおりです．

特徴 / 入力方式	概　要	日本語入力の簡便さ	メモリ/ファイル所要量
連文節変換入力方式	“読み”→“日本語”の変換を複数の文節に渡って一度におこないます．ただし，利用者メモリの圧迫が単文節変換入力方式よりも大きくなります（単文節変換入力方式の66KB[注]にくらべ114KB[注]のメモリが必要です）．	簡単 ↑	大きい ↑
単文節変換入力方式	“読み”→“日本語”の変換を文節単位におこないます．連文節変換入力方式にくらべ利用者メモリの圧迫が小さくてすみます（66KB[注]必要です）．		
JISコード入力方式	単字単位に対応する日本語のJISコードを入力していきます．利用者メモリを圧迫しません．辞書ファイルも使用しません．	↓ めんどう	↓ 小さい

注：基本システムコード24KBを含んでいます．

いずれの入力方式で入力できるかはシステムの起動時に決定されます．

1．連文節変換入力

(1) 連文節変換入力用システムディスクを使用します．システムディスクは連文節変換入力用に特殊化された状態で出荷されます．

従って，システムディスクを標準の状態で使用しますと，日本語入力の際，連文節変換入力が行えます．

(2) 連文節変換の際，使用される辞書ファイルは，“BUNSETSU”という名前でシステムディスクに格納されています．この辞書ファイルは単文節変換入力環境においても使用されます．

(3) “読み”を日本語の並びに変換する際，システムは辞書ファイルを参照しますが，この時，次の規則に従って辞書ファイルを探します．

① 標準のシステムディスクを使用する場合（出荷時の状態で使用する場合），システム起動時にシステムディスクがセットされていたドライブに対し辞書ファイルを探しにいきます．

② 辞書ファイルのドライブ番号を固定的に宣言しておくことができます．

“setup.n88”ユーティリティで辞書ファイルが存在するドライブ番号をシステムディスクに与えます．

このシステムディスクでシステムを起動しますと，指定されたドライブに対して固定的に辞書ファイルを探しにいきます．

③ 固定ディスクからシステムを起動する場合，固定ディスクに設定されたDISK CODEの属性によって辞書ファイルのドライブが決まります．

・DISK CODEが標準のシステムディスクから移送されたものである場合，システムが起動された固定ディスクに対して辞書ファイルを探しにいきます．

・DISK CODEが辞書ファイルのドライブ番号を固定的に宣言したシステムディスクから移送されたものである場合，固定ディスクから起動した場合でも，指定されたドライブに対して固定的に辞書ファイルを探しにいきます．

④ 上記のいずれの場合でも，辞書ファイルのドライブ番号を日本語入力モードに入った後変更することができます．

辞書の切り換え機能（f·7 キーによる）を使用し，辞書ファイルの存在するドライブ番号を宣言します（この時，辞書ファイル名も同時に指定します）．

ここで変更した辞書ファイルの環境はシステムを終了するまで，あるいは更に辞書の切り換えを行うまで有効です．

(4) 辞書ファイルの名前は自由に変更することができます（NAMEコマンドで変更して下さい．NAMEコマンドについては「BASICリファレンスマニュアル」を参照して下さい）．

標準名“BUNSETSU”以外の名前の辞書ファイルを使用する場合，日本語入力モードに入った後（KINPUT文実行後あるいは CTRL ＋ XFER 入力後），辞書の切り換え機能（f·7 キーによる）を使用し，辞書ファイル名を宣言する必要があります．

(5) 単文節変換入力あるいはJISコード入力方式用に特殊化したシステムディスクを連文節変換入力用システムディスクに変更することができます．

“setup.n88”ユーティリティを使用し，システムディスクの日本語入力の方法を規定します．

備　考

“setup.n88” ユーティリティはシステムディスクに次の属性を与えます．

(1) そのシステムディスクで取り扱うことができる日本語入力（連文節変換入力，単文節変換入力，JISコード入力のいずれか）の方法

(2) 辞書ファイルのドライブ番号

“setup.n88” ユーティリティの詳細については「第15章　システムディスクに格納されているユーティリティプログラム」を参照して下さい．

2．単文節変換入力方式

(1) 単文節変換入力用システムディスクを使用します．標準のシステムディスクは連文節変換入力用に特殊化されていますので，このままでは単文節変換入力環境での日本語入力は行えません．

“setup.n88” ユーティリティを使用し，システムディスクを単文節変換入力用に特殊化します．

(2) 辞書ファイルの名前や辞書ファイルの探索についての規則は連文節変換入力環境の場合と同一です．

3．JISコード入力方式

(1) JISコード入力用システムディスクを使用します．単文節変換入力用システムディスクの場合と同様に“setup.n88” ユーティリティを使用し，システムディスクをJISコード入力用に特殊化します．

(2) JISコード入力方式の日本語入力を行う場合，辞書ファイルは必要としません．

備　考

ROMモードBASICにおいては，JISコード入力方式しか行うことはできません．

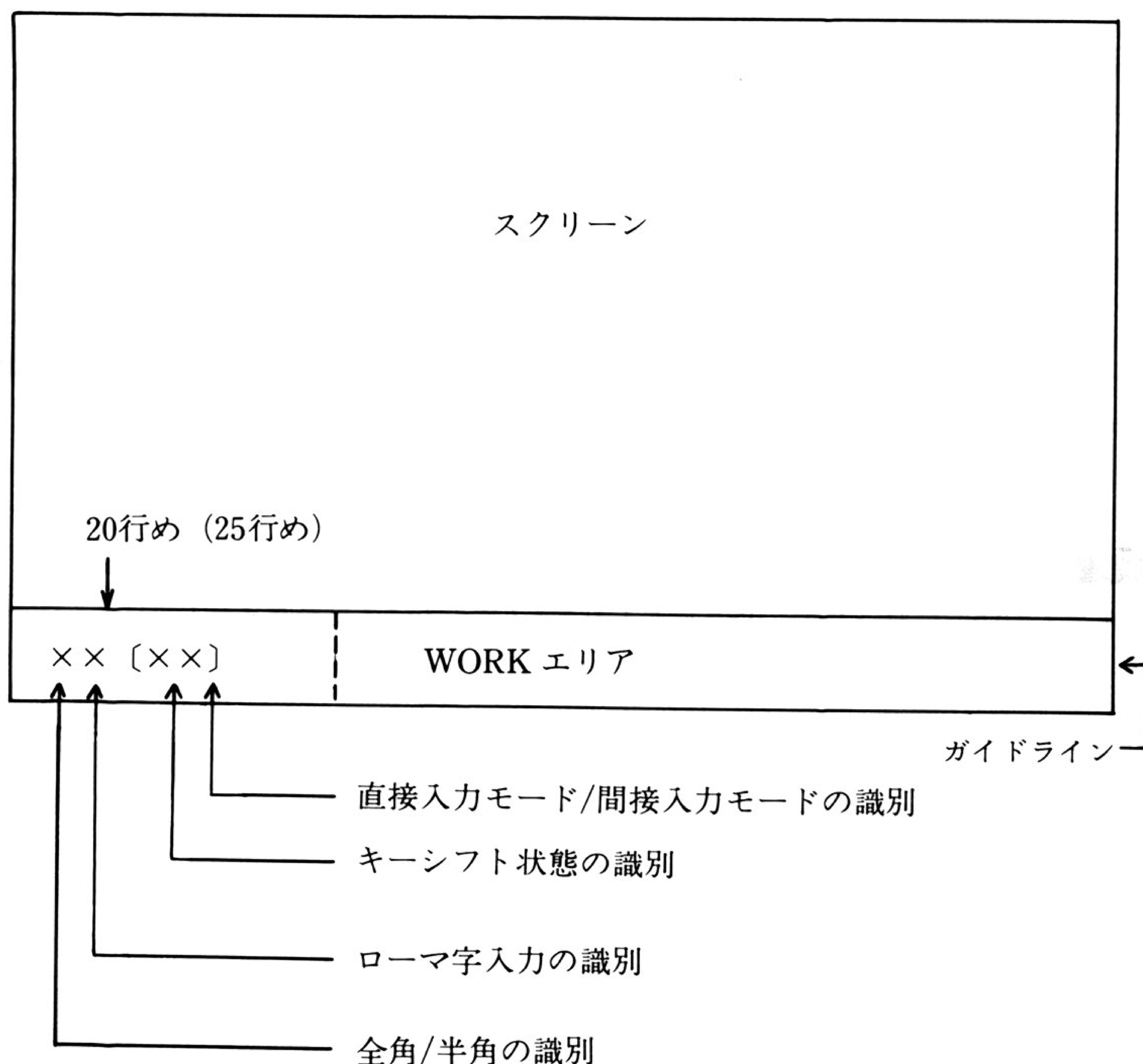
5.1 連文節・単文節変換入力方式

連文節変換入力環境と単文節変換入力環境における操作法や種々の規則は，“読み”を“日本語”に変換する際の一括変換能力の違いをのぞき基本的に同一です．特に断り書きがない限り，連文節入力環境と単文節入力環境で共通の操作あるいは規則であると御理解下さい．

5.1.1 入力モードの表示

日本語入力モードに入ると，画面最下行に次のようなガイドラインが表示されます．(20行モードのときは20行め，25行モードのときは25行めに表示されます.)

注 意　40桁モードの場合，文節変換入力方式の日本語入力はおこなえません．JISコード入力方式による日本語入力となります．



(1) ガイドラインの説明

① 全角/半角の識別

*……半角

キーボードから入力された文字を半角として扱います．ただし，ひらがな入力での半角は，半角カタカナとなります．

空白…全角

キーボードから入力された文字を全角として扱います．

A……1バイトコード

キーボードから入力された文字を1バイト系の文字として扱います.

② ローマ字入力の識別

R……ローマ字

ローマ字をキーボードから入力して，カタカナまたはひらがなの入力ができる状態です.

空白…英数字またはカナ

英数字の入力ができる状態です. [カナ] キーをロックしてカタカナ/ひらがなを入力する場合は，この状態です.

③ キーシフト状態の識別

英数…英数字，英記号

英文字，数字，英記号の入力ができる状態です.

カナ…カタカナ

ローマ字またはカナで，カタカナの入力ができる状態です.

かな…ひらがな

ローマ字またはカナで，ひらがなの入力ができる状態です.

JIS… JIS16進コード

JIS16進コードの入力ができる状態です.

設定…辞書の先読みまたは学習機能設定
(学習)

辞書の先読みや学習機能が選択できる状態です(単文節変換の場合,学習機能のみ).

情報…情報表示

辞書ファイル名，学習機能などを表示する状態です.

④ 直接入力モード/間接入力モードの識別

〔　〕…直接入力モード

キーボードから入力された文字が，そのまま現在カーソルのある位置に表示されます. その位置で漢字に変換することもできます.

【　】…間接入力モード

キーボードから入力された文字が，まずガイドラインに表示されます.

そのまま現在カーソルのある位置に入力するか，または漢字に変換して入力することができます.

⑤ WORKエリア

モード状態によって使用方法が異なりますので，表示内容は後述の各モードの説明を参照してください.

また，このエリアはシステムのエラーメッセージ表示にも使用されています.

5.1.2 入力モードの切り換え

(1) 直接入力モード/間接入力モードの切り換え ([f·1] キー)

文節変換入力モードに入ると，まず直接入力モードになります. 直接入力モードと間接入力モードの切り換えは，[f·1] キーで行います. ただし，直接入力モードから間接入力モードに切り換える時，辞書ファイルがないとブザーが鳴り，モードは切り換わりません.

・**直接入力モード**



このモードでは，キーボードから入力された文字をカーソルのある位置にリバース（反転）表示します．漢字変換もこの位置で行います．



間接入力に切り換えるには，読みが入力されていない時に f·1 キーを押してください．

・**間接入力モード**



このモードでは，キーボードから入力された文字を一時的に画面最下行のガイドラインに表示します．漢字変換を行う場合には，このガイドライン上で変換してから，カーソル位置に移します．



直接入力に切り換えるには読みが入力されていない時に f·1 キーを押してください．

(2) キーシフト状態の切り換え

① カナ文字入力/英数字入力の切り換え（ カナ キー）

カナ キーのロック状態によって入力文字を変えることができます．

・ カナ キーがロックされている状態

カナ文字によるカタカナ，ひらがなの入力ができます．

カ ナ とキーインすると次のように表示されます．



・ カナ キーがロックされていない状態

英数字による入力で英数字やローマ字によるカタカナ/ひらがなの入力ができます．

K A N A とキーインすると，次のように表示されます．

```
かな■
  R〔かな〕
```

② 英数/ローマ字入力の切り換え（[f·2]キー）

カナキーがロックされていない状態で，英数字を入力するか，ローマ字によってカタカナ/ひらがなを入力するかを切り換えます．この切り換えは[f·2]キーで行います．

・**英数入力**

[A][B][C]とキーインすると次のように表示されます．

```
ABC■
  〔英数〕
```

ローマ字入力に切り換えるには[f·2]キーを押してください．

・**ローマ字入力**

ローマ字によってカタカナまたはひらがなを入力します．

[A][I][U]とキーインすると次のように表示されます．

```
アイウ■
  R〔カナ〕
```

英数入力に切り換えるには[f·2]キーを押してください．

③ カタカナ/ひらがな入力の切り換え（[f·3]キー）

入力されたカナ文字をカタカナに変換するか，ひらがなに変換するかを切り換えます．この切り換えは[f·3]キーで行います．

・**カタカナ入力**

入力された文字をカタカナに変換します．

[K][A][N][A]とキーインすると次のように表示されます．

```
カナ■
  R〔カナ〕
```

ひらがな入力に切り換えるには[f·3]キーを押してください．

・**ひらがな入力**

入力された文字をひらがなに変換します．

[K][A][N][A]とキーインすると次のように表示されます．

```
かな■
  R〔かな〕
```

カタカナに切り換えるには f·3 キーを押してください．

④ 全角/半角入力の切り換え（f·4 キー）

キーボードから入力された文字を全角文字に変換するか半角文字に変換するかを切り換えます．この切り換えは f·4 キーで行います．

・全角入力

キーボードからの入力を全角文字に変換します．

初期にはこの状態になっています．

KANAとキーインすると次のように表示されます．

```
カナ■
  R〔カナ〕
```

半角入力に切り換えるには読みが入力されていない時に f·4 キーを押してください．

・半角入力

キーボードからの入力を半角文字に変換します．

日本語入力で扱う半角文字は2バイトコードですが，スペース（空白）だけは1バイトの20Hとなるので注意してください．

KANAとキーインすると，次のように表示されます．

```
カナ■
  *R〔カナ〕
```

全角入力に切り換えるには読みが入力されていない時に f·4 キーを押してください．

⑤ JIS16進コード入力モード（f·5 キー）

特殊記号など，キーボードから入力できない文字や，辞書に登録されていない文字などを入力する場合に使用するもので読みが入力されていない時に f·5 キーを押すと，1文字単位でJISコードの16進表記4桁で入力できます．

このとき直接入力モードになり，間接入力することはできません．1文字の入力が終わるとその場で確定し（リバースされません）自動的にもとの入力モードに戻ります．

```
■
  〔JIS〕
```

2178とキーインすると次のように表示されます．

```
§■
  〔英数〕
```

このモードから抜けたい時は，任意の時点で f·5 キーを押してください．もとの入力モードに戻ります．

また，このモードから，1バイトコード入力，辞書の切り換え，単語登録，単語削除，情報表示，辞書の先読みあるいは学習機能の設定へ入ることはできません．

⑥　1バイトコード入力モード（ f·6 キー）

キーボードから入力された文字を1バイトコードで入力するか，2バイトコードで入力するかを切り換えます．この切り換えは読みが入力されていない時に f·6 キーで行います．

・1バイトコード入力モード

直接入力となり，1バイトコードの入力が可能となります．

A S A とキーインすると次のように表示されます．

```
アサ■
  AR〔カナ〕
```

2バイトコード入力モードに切り換えるには f·6 キーを押してください．

また，このモードからJIS16進入力，辞書の切り換え，単語登録，単語削除，情報表示，辞書の先読みあるいは学習機能へ入ることはできません．

5.1.3　文字の入力

英数字，英記号の入力

英数字・英記号は，ガイドラインのシフト表示が“英数”になっている状態で入力できます．

通常の1バイトコードと同様， CAPS キー， SHIFT キーを使って大文字，小文字，英記号を入力します．

例　「ABCD」の入力

A B C D とキーインします．

直接入力モードでは，カーソルの位置にそのまま入力されます．

```
ABCD■
  〔英数〕
```

間接入力モードでは，ガイドライン上の読み入力フィールドに表示されます．

```
■
  【英数】ABCD■                                             〈
```

(1)　全角文字

ガイドラインのシフト表示が“英数”になっている時，キーボードの英数字あるいは英記号のキーを押すことにより，全角の英数字（大文字，小文字）および英記号が入力できます．

(2) 半角文字

ガイドラインのシフト表示が“英数”で，その左側に“＊”が表示されている時に，半角の英数，英記号が入力できます．

カタカナ/ひらがなの入力

カナ文字の入力には，キーボードからカナ文字を入力する方法と，ローマ字で入力したものをカナに変換する方法があります．

(1) 半角の扱い

カナ入力状態にある時，[f·4]キーを押すと半角カナ入力となり，ガイドライン左端に“＊”が表示されます．

この場合，ガイドラインのシフト表示が“カナ”になり，入力文字はカタカナに変換されます．また，濁点や半濁点はそのまま1文字として扱います．

半角カナ入力では，[f·3]キー（カタカナ/ひらがな入力の切り換え）は無視されます．

(2) カナ文字による入力

ガイドラインのシフト表示が“英数”になっているとき，[カナ]キーをロックすることにより，シフト表示がかな（カナ）になり，カナ文字によるひらがな，またはカタカナの入力ができるようになります．

・カナ文字による入力での濁音，半濁音の扱い

濁音，半濁音は[カ]＋[゛]のようにカナ文字の後で濁点，または半濁点を入力します．

例

先程の英数字に続けて「かんじ」と入力してみます．

[カナ]キー，[f·3]キーを操作してカナ文字によるひらがな入力の状態にします．

[カ][ン][シ][゛]とキーインします．

画面には次のように表示されます．

直接入力モード

```
ＡＢＣＤかんじ■
  〔かな〕
```

間接入力モード

```
■
  【かな】ABCDかんじ                                  〈
```

(3) ローマ字による入力

ガイドラインのシフト表示が“英数”になっているとき，[f·2]キーを押すことによりローマ字によるカタカナまたはひらがなの入力ができるようになります．

例

ローマ字で，「かんじ」と入力してみます．

f·2，f·3 キーを操作してローマ字によるひらがな入力の状態にします．

KANJI とキーインします．

画面には次のように表示されます．

直接入力モード

ＡＢＣＤかんじ■

R〔かな〕

間接入力モード



R【かな】ＡＢＣＤかんじ■

ローマ字によるカナ文字の入力は，次の規則に従って行っています．

ア	イ	ウ	エ	オ
a	i	u	e	o
カ	キ	ク	ケ	コ
ka	ki	ku	ke	ko
サ	シ	ス	セ	ソ
sa	si (shi)	su	se	so
タ	チ	ツ	テ	ト
ta	ti (chi)	tu (tsu)	te	to
ナ	ニ	ヌ	ネ	ノ
na	ni	nu	ne	no
ハ	ヒ	フ	ヘ	ホ
ha	hi	hu (fu)	he	ho
マ	ミ	ム	メ	モ
ma	mi	mu	me	mo

ヤ		ユ		ヨ
ya		yu		yo
ラ	リ	ル	レ	ロ
ra	ri	ru	re	ro
ワ				ヲ
wa				wo
ガ	ギ	グ	ゲ	ゴ
ga	gi	gu	ge	go
ザ	ジ	ズ	ゼ	ゾ
za	zi (ji)	zu	ze	zo
ダ	ヂ	ヅ	デ	ド
da	di	du	de	do
バ	ビ	ブ	ベ	ボ
ba	bi	bu	be	bo
パ	ピ	プ	ペ	ポ
pa	pi	pu	pe	po

キャ	キュ	キョ
kya	kyu	kyo
シャ	シュ	ショ
sya (sha)	syu (shu)	syo (sho)
チャ	チュ	チョ
tya (cha)	tyu (chu)	tyo (cho)
ニャ	ニュ	ニョ
nya	nyu	nyo
ヒャ	ヒュ	ヒョ
hya	hyu	hyo
ミャ	ミュ	ミョ
mya	myu	myo

リャ	リュ	リョ
rya	ryu	ryo
ギャ	ギュ	ギョ
gya	gyu	gyo
ジャ	ジュ	ジョ
zya (ja)	zyu	zyo (jo)
ヂャ	ヂュ	ヂョ
dya	dyu	dyo
ビャ	ビュ	ビョ
bya	byu	byo
ピャ	ピュ	ピョ
pya	pyu	pyo

備　考

Ⓐ 長音記号

[^へ] または [¥]

Ⓑ 促音（っ）

ⓐ 子音を重ねる　itta→イッタ

ⓑ 母音，[SHIFT] ＋ [7ャ]　a’→アッ

Ⓒ ン，ん

ⓐ nの次に子音　kanji→カンジ

ⓑ n，[SHIFT] ＋ [7ャ]　kan’i→カンイ

ⓒ nの次にyがある場合はnya→ニャ

nyu→ニュ

nyo→ニョ

Ⓓ ヲ，を

wo

Ⓔ カナ文字の場合の特殊入力

ファ	フィ	フェ	フォ	ジェ	シェ
fa	fi	fe	fo	je	she
チェ	ヴァ	ヴィ	ヴ	ヴェ	ヴォ
che	va	vi	vu	ve	vo

Ⓕ カナ小文字の入力

[SHIFT] ＋a, i, u, e, o

Ⓖ 読点

[SHIFT] ＋ [> 。ル]

Ⓗ　句点

[SHIFT] ＋ [く]

1バイトコードの入力

1バイトコードの入力は，ガイドラインのモードに従って入力することができます．
英数字の入力は，通常の1バイトコードの入力と同様です．
カナの入力は，キーボードからカナ文字を入力する方法と，ローマ字で入力したものをカナに変換する方法があります．

1バイトコードの入力は直接入力のみで，入力された文字はカーソル位置に表示され，その場で確定します．従って読みとして使うことはできません．

(1)　英数字の入力

ガイドラインのシフト状態表示が“英数”になっている時[f･6]キーを押すと，ガイドラインの先頭に“A”が表示され1バイトコードの英数字の入力ができます．

例　ABCDと入力してみます．

[f･2]キー，[f･6]キーを操作して1バイトコードの英数字入力の状態にします．

[A][B][C][D]とキーインします．

画面には次のように表示されます．

```
ABCD■
 A〔英数〕
```

(2)　カタカナの入力

ガイドラインの先頭が“A”になっているとき（[f･6]キーを押し）[カナ]キーをロックすると，カナ文字による，1バイトコードのカタカナの入力ができます．

また，ガイドラインの先頭が[f･6]キーにより“A”になっているとき[f･2]キーを押すと，ローマ字による1バイトコードのカタカナの入力ができます．

例

先程の英数字に続けて「アイウ」と入力してみます．

・カナ文字による入力

[カナ]キー，[f･6]キーを操作して，カナ文字によるカタカナ入力の状態にします．

[ア][イ][ウ]とキーインします．

画面には次のように表示されます．

```
ABCDアイウ■
 A〔カナ〕
```

・ローマ字による入力

[f･2]キー，[f･6]キーを操作してローマ字によるカタカナ入力の状態にします．

[A][I][U]とキーインします．

画面には次のように表示されます．

```
ABCDアイウ■
  AR〔カナ〕
```

漢字の入力

(1) 読みの入力

① まず漢字の読みを入力します．

・直接入力モード

```
かんじが■
  〔かな〕
```

・間接入力モード

```
■
  【かな】  かんじが■                                  〈
```

上図の“＜”はストッパーです．入力できる文字の目安として表示してあります．

② 入力した読みが間違っていた場合は，下記の各キーを使用して，読みを修正してください．

・ BS キー， ← キー
カーソル（間接入力モードでは擬似カーソル）を1文字分左へ移動します．

・ → キー
カーソル（間接入力モードでは擬似カーソル）を1文字分右へ移動します．

・ ESC キー
読みをすべて消去します．

・ DEL キー
カーソル（間接入力モードでは擬似カーソル）の左側の1文字を消去します．

・ INS キー
挿入モードに入ります．
それ以後カーソル（間接入力モードでは擬似カーソル）の位置に入力文字が挿入されます．

・ TAB キー
カーソル（間接入力モードでは擬似カーソル）を読みの最後の文字の次に移動します．

・ HOME CLR キー
カーソル（間接入力モードでは擬似カーソル）以降の読みを消去します．

・ SHIFT ＋ HOME CLR キー
カーソル（間接入力モードでは擬似カーソル）を読みの先頭に移動します．

上記の各キーの使用例は「5.1.5キーの使用」で解説しています．

(2)　漢字の変換

連文節変換

連文節変換は，読みを入力した後 [XFER] キーを押して行います．

読みは32文字まで入力できます．ただし一度に処理できるのは16文節までと決まっています．

・**直接入力**

①　読みを入力した後 [XFER] キーを押すと，読みがその位置で漢字まじりの文章に変換されて表示されます．このとき先頭の文節が白地反転され，次候補などの操作の対象となります．後続の文章は，下線つきで表示されます．

> わたしはきょういしゃへいきます
> 〔かな〕

ここで [XFER] キーを押すと，たとえば次のように変換されます．

> 私は京井者へ行きます
> 〔かな〕

ここで [ESC] キーを押すと白地反転の文節がもとの読みに戻ります．

> わたしは京井者へ行きます
> 〔かな〕

ここでもう一度 [ESC] キーを押すと全文章が読みに戻ります．

②　白地反転された文節が希望のものでないとき，再び [XFER] キー（または [↓] キー）を押して再変換を行います．

> 渡しは京井者へ行きます
> 〔かな〕

誤って希望の候補を通り過ぎてしまったときは [↑] キーで前の候補に戻ることができます．

③　[XFER] キー，[↓] キーで候補を表示させているとき候補がなくなると，もとの読みが表示され，カーソルが読みの最後の文字の次に表示されます．

> わたしは京井者へ行きます
> 〔かな〕

④　表示されている候補が希望のものであれば，その時点でリターンキーを押します．これで先頭の文節が確定され，白地反転が次の文節に移動します．

私は京井者へ行きます
　　〔かな〕

このような操作を繰り返して読みを希望の文章に変換してください．

もし，白地反転されている候補と下線表示されている後続の文章がすべて希望のものと一致していれば，次の文章の読み入力を行うかスペースキーを押してください．

文章全体を一度に確定させることができます．

私は今日医者へ行きます
　　〔かな〕

スペースキーを押すと

私は今日医者へ行きます■
　　〔かな〕

注　意　この場合のスペースキーは空白の入力にはなりません．

連文節変換においては，変換対象の文節の読みをなるべく長くとるように変換を行います．文節の区切りが希望どうりでないときでも，[XFER] キー，[↓] キーによって候補を表示していくと，だんだん文節の読みを短くして変換していきますので，何度めかの候補で希望の文になるはずです．

例　京井（きょうい）
　　↓
　　今日（きょう）
　　↓
　　巨　（きょ）
　　↓
　　木　（き）

対象文節の読みが変わるような変換が行われた場合，後続の文章を変換しなおして画面に表示します．

私は京井者へ行きます　　下線部の読みは
　　〔かな〕　　　　　　　「しゃへいきます」

↓　何回か [XFER] キーを押す

私は今日医者へ行きます	下線部の読みは
〔かな〕	「いしゃへいきます」となる

注　意　(1)　読みに句読点（“，”や“。”）を入れる場合には，句読点ごとに変換操作を行なってください．

[ト][ウ][ロ][ク][ス][ル][ト][、][シ][゛][シ][ョ][ノ]　[XFER]　…×

[ト][ウ][ロ][ク][ス][ル][ト][、]　[XFER]　…○

(2)　一度 [XFER] キーで変換を行なった文章については，その文章全体を確定させてから次の文章の読みの入力を行なってください．

(3)　[XFER] キーによる最初の候補は適当と思われる長さの読みで変換されたものが表示され，２度めの [XFER] キー以降，読みの長い順に表示されますので，最初の候補と２番め以降の候補で同じものが表示されることもあります．

・間接入力

間接入力は，読みと候補とがともにガイドラインに表示されることを除けば，直接入力と同じです．

ただし，読みは白地反転されません．

■
【かな】　わたしはきょういしゃへいきます■　く

↓　[XFER] キー

■
【かな】　私は京井者へ行きます

↓　リターンキー

私は■
【かな】　京井者へ行きます

↓　[XFER] キー（何回か押す）

私は■
【かな】　今日医者へ行きます

↓　次の文章の入力を開始する．

私は今日医者へ行きます■
【かな】　の■　　　　　　　　　　　　　　　　　　　　〈

・**表示選択**（[SHIFT]＋[XFER]キー）

[SHIFT]＋[XFER]キーで表示選択モードに入ります．

① 読みを入力し，[SHIFT]＋[XFER]キーを押すと，読みを入力した位置に全文章の候補が表示され，反転表示されている文節に対する候補がガイドラインに最大9つまで表示されます．

京井者へいきます
〔かな〕　1 京井　2 脅威　3 驚異…

② ガイドライン上の候補群の中に希望の候補がない場合，[XFER]キー，[↑]キーや[↓]キーで候補群の表示を切替えます．表示された候補群の中に希望の候補が見つかったら，[←]キー，[→]キーで擬似カーソルを移動するか，候補の番号を入力してその候補に擬似カーソルをかさねます．

京井者へ行きます
〔かな〕　…　6 今日　7 京…

読みを入力した位置に表示されている候補は，[SHIFT]＋[XFER]キーを押した時点のものがそのまま表示されています．

③ リターンキーを押すと，カーソル位置の候補が選択され，表示選択モードから抜けます．

今日医者へ行きます
〔かな〕

注　意　表示選択モードに入ると，スペースキーや次の読みの入力によって全文章を確定させることはできません．全文の確定は通常のモードでおこなってください．

単文節変換

漢字の変換は読みを入力した後[XFER]キーを押して行います．

読みは32文字まで入力できます．ただし，一度に変換できるのは16文字までです．

カタカナ，カナ記号（“ー”，“「”，“」”，“，”，“．”，“・”），“！”，“*”，および“　”（スペース）は変換できません．

また，読みの中に全角と半角を混在させることはできません．

・**直接入力**

① 読みを入力した後，[XFER]キーを押すと，読みがその位置で漢字に変換されて反転（リバース）表示されます．この時カーソルは表示されません．

幹事が
〔かな〕

② XFER キーを押しても辞書にその読みの漢字がない場合には，もとの読みが表示され，カーソルが読みの先頭に移動します．

ESC キーを押して読みを入力しなおすか，カーソル移動キーで読みを修正してください．

かんぎが
〔かな〕

③ 表示された漢字が希望の漢字でない場合は XFER キーまたは ↓ キーを押すと，次の候補が表示されます．誤って希望の候補を通りすぎてしまった場合，↑ キーを押すと前の候補に戻ります．

漢字が
〔かな〕

④ XFER ，↓ キーを押して候補を表示させている時，表示する候補がなくなると，もとの読みが表示され，カーソルが読みの最後の文字の次に移動します．

かんじが■
〔かな〕

⑤ 表示されている候補が希望の漢字であればその時点でリターンキーを押すか，次の読みを入力します．

変換された文字の反転（リバース）が消え，カーソルが最後の文字の次に移動します．

これで漢字変換の終了です．

漢字が■
〔かな〕

⑥ XFER キーを押した後に読みの打ち間違いに気付いた場合は，ESC キーを押すと読みの入力に戻ります．ここで(1)読みの入力で示した各キーを使用して読みの修正を行い再び漢字変換を行ってください．

間接入力

間接入力は変換された文字がガイドラインに表示されますが，その他の操作は直接入力と同じです．但しガイドライン上の文字は反転（リバース）しません．

① 読みを入力します．

```
■
　【かな】かんじが■　　　　　　　　　　　　　　　　　　　　〈
```

② XFER キーを押して変換します．この時，擬似カーソルとストッパーは表示されません．

```
■
　【かな】幹事が
```

③ XFER キーまたは ↓ キーを押して次の候補を表示していきます．前の候補を表示するには ↑ キーを押します．

```

　【かな】漢字が
```

④ 希望の漢字が表示されたらその時点でリターンキーを押すか次の読みを入力します．
変換された文字がガイドラインから消えて現在のカーソル位置に移動します．
これで漢字変換の終了です．

```
漢字が■
　【かな】■　　　　　　　　　　　　　　　　　　　　　　　〈
```

⑤ 読みの修正を行う時は ESC キーを押して読みの入力に戻してから，(1)読みの入力 で示したキーを使用して修正してください．

・**表示選択（ SHIFT ＋ XFER キー）**

SHIFT ＋ XFER キーで表示選択モードに入ります．
直接入力，間接入力，部首選択のどれでもこの機能を利用することができます．
表示を読みに戻すか，候補を選択すると，通常のモードに戻ります．

① 読みを入力します．

```
ききます■
　〔かな〕
```

② SHIFT ＋ XFER キーを押すとガイドラインに数種類の漢字候補が表示され，先頭の漢字には擬似カーソルが重ねられます．候補は最大 9 つまで可能なだけ表示されます．候補表示中にこのモードに入った場合は，その次からの候補群を表示します．

```
利きます
　〔かな〕　　　　　　　1 利きます 2 聞きます……
```

③　表示された候補群の中に希望の漢字がない場合は XFER または ↓ キーを押すと次の候補群が表示されます．前の候補群を表示するには ↑ キーを押します．

表示する候補群がなくなると，画面は，そのまま最後の候補群を表示します．

```
効きます
　〔かな〕　　　　　1 効きます　2 聴きます……
```

④　表示されている候補群の中に希望の漢字があれば，→ キー，← キーを使って希望の漢字に擬似カーソルを移動します．

```
聞きます
　〔かな〕　　　　　1 利きます　2 聞きます……
```

⑤　リターンキーを押すことにより，カーソル位置の候補が選択されます．

また，希望の漢字の番号を入力して直接その候補を選択することもできます．

これで漢字変換の終了です．

```
聞きます■
　〔かな〕
```

⑥　読みの修正を行うときは，ESC キーを押して読みの入力に戻してから，(1)読みの入力　で示した各キーを使用して修正してください．

・**連続変換**

前述の変換を連続的に行います．

連続変換は複数の文節の読みを先に入力しておき，後でまとめて変換（無変換）操作を行うものです．

連続して変換するための読みは31文字までで，変換されるのは，読みとして入力された反転（リバース）表示の部分だけです．間接入力の場合はガイドラインに表示された読みが変換されます．

①　読みを入力します．

```
れんぞくしてへんかんします■
　〔かな〕
```

②　カーソル移動キーで変換したい語句の最後の文字の次にカーソルを移動させます．

```
れんぞくしてへんかんします
　〔かな〕
```

③ XFER キーを押します．

変換したい部分がその位置で変換されて表示されます．読みの残りの部分とカーソルはこの時表示されません．

表示された漢字が希望の漢字でない場合は XFER キーまたは ↓ キーで次の候補，↑ キーで前の候補を表示します．

```
連続して
  〔かな〕
```

④ 辞書にその読みがない場合は，もとの読みがすべて表示され，カーソルが読みの先頭になります．

```
れんぞくしてへんかんします
  〔かな〕
```

⑤ 変換して，それ以上候補のない場合にはもとの読みがすべて表示され，カーソルが変換していた語句の最後の文字の次に移動します．

```
れんぞくしてへんかんします
  〔かな〕
```

⑥ 希望の漢字が表示されたら，その時点でリターンキーを押します．

・直接入力の場合は，変換された文字の反転（リバース）表示が消え，もとの読みの残りの部分が，反転表示されます．この時カーソルは反転部分の最後に移動します．

```
連続してへんかんします■
  〔かな〕
```

・間接入力の場合は，変換された文字がガイドラインから消え，現在のカーソルの位置に入力されます．もとの読みの残りの部分がガイドラインに表示され，擬似カーソルが読みの最後に移動します．

```
連続して■
  【かな】へんかんします■
```

⑦ 以降は②からの操作をくり返してください．

```
連続してへんかんします■
  〔かな〕
```

XFER キー（数回）

```
連続して変換します■
  〔かな〕
```

リターンキー

```
連続して変換します■
  〔かな〕
```

⑧　これで入力が終わりました.

⑨　連続変換で読みの修正を行う時も ESC キーを押して読みの入力に戻してから，(1)読みの入力で示した各キーを使用して修正してください.

直接入力の時のガイドライン直前での入力

直接入力の時，ガイドライン直前の行からガイドラインへかけて入力する時は，一時的に間接入力モードに切り換わります.

```
                        現在は，画面の一番下の行に入力しています．つ
  R〔かな〕
```

読みを入力し続けると間接入力モードになり次のようになります.

```
                          現在は，画面の一番下の行に入力しています．
  R【かな】つぎの，■                                                〈
```

間接入力モードで変換操作を行います.

```
                          現在は，画面の一番下の行に入力しています．
  R【かな】次の，                                                    〈
```

リターンキーが押され，文字が確定すると画面がスクロールして，直接入力モードに戻ります.

```
                        現在は，画面の一番下の行に入力しています．次
の，■
    R〔かな〕
```

最下位行での入力

最下位行では直接入力しかおこなえません.

一度，最下位行での入力をおこないますと，最下位行以外の行で入力をおこなう場合でも，

CTRL ＋ f·6 キーを押すまでガイドラインは表示されません．

郵便番号による変換

読みの入力に対し，数字３桁を入力し変換をおこないますと，数字３桁は郵便番号と解釈され，該当する住所に変換されます．

5.1.4 各種機能

辞書の切り換え（f·7 キー）

f·7 キーを押すと，以下の画面が表示され，辞書の切り換え指定が行えます．
読みや，候補の表示中は，この機能は使えません．



現在の辞書ファイル名を表示して，入力待ちになります．

辞書ファイル名の形式は以下のようになっています．

〔d：〕ファイル名　〔　〕内は省略可能

d：　ドライブ番号指定
省略するとシステム起動時に決定されたドライブになります．

ファイル名　一般のディスクファイル名の入力規則にしたがって入力します（漢字やかたかなは使用できません）．

ファイル名を入力しリターンキーを押すと，辞書ファイルを切り換えて直前のモードに戻ります．

必らず文節入力用の辞書を指定してください．

このモードから抜けるためには，読みのない状態で ESC キーを押してください．直前のモードに戻ります．

このモードから単語登録，単語削除，情報表示，辞書の先読みあるいは学習機能の設定モードへ入ることはできません．

単語登録（f·8 キー）

f·8 キーを押すと以下の画面が表示され，単語登録のモードに入ります．
このモードでは，画面上の文字を辞書に登録することができます．



例

“NEC”という読みで“日本電気㈱”を登録してみます．

① カーソル移動キーで，カーソルを画面上の登録する語句の先頭に移動します．そこでリターンキーを押します．指定された文字が漢字の場合，反転（リバース）となります．

```
日本電気㈱
AR〔カナ〕　登録：漢字
```

② カーソルを，登録する語句の最後の文字に移動してリターンキーを押します．単語の先頭から最後の文字まで全て漢字で，16文字以内の場合は，その単語が反転し，ガイドラインが次のように変わります．

```
日本電気㈱
  AR〔カナ〕　登録：読み　■
```

③ ガイドライン上に1バイトコード入力（ガイドラインの左端がA）で読みを入力します．ただし，読みは16文字以内です．

```
日本電気㈱
  AR〔英数〕　登録：読み　NEC■
```

④ 読みを入力後リターンキーを押すと，辞書に単語を登録します．これで登録が終了し直前のモードに戻ります．

```
日本電気㈱■
R〔かな〕
```

このモードから抜けるためには，任意の時点で ESC キーを押してください．直前のモードに戻ります．このモードで登録した単語は，ここで指定する“読み”でのみ変換が可能になります．

このモードからJIS16進入力，辞書の切り換え，単語削除，情報表示，辞書の先読みあるいは学習機能の設定モードへ入ることはできません．

注 意 辞書に登録されていない“読み”に対して単語登録をおこなうことはできません．新しい“読み”に対する単語登録は“dicmen.n88”ユーティリティでおこなってください．

単語削除（ f·9 キー）

f·9 キーを押すと，以下の画面が表示され，単語削除のモードに入ります．
このモードでは，画面上の文字を辞書から削除することができます．
ただし，削除できる単語は利用者が登録した単語だけです．

```
日本電気㈱■
AR〔カナ〕　削除：漢字
```

例

“NEC”という読みの“日本電気㈱”を削除してみます．

操作方法は登録の場合と同じです．

① カーソル移動キーでカーソルを画面上の削除する単語の先頭に移動します．そこでリターンキーを押します．指定された文字が漢字（2バイトコード）の場合，反転（リバース）表示されます．

```
日本電気㈱
AR〔カナ〕　削除：漢字
```

② カーソルを削除する単語の最後の文字に移動してリターンキーを押します．単語の先頭から最後の文字まですべて漢字（2バイトコード）で16文字以内の場合は，その単語が反転し，ガイドラインが次のように変わります．

```
日本電気㈱
AR〔カナ〕　削除：読み　■
```

③ ガイドライン上に1バイトコード入力（ガイドラインの左端が“A”）により読みを入力します．

ただし読みは16文字以内です．

```
日本電気㈱
AR〔英数〕　削除：読み　NEC■
```

④ リターンキーを押すと辞書から単語が削除されます．これで削除が終了し，直前のモードに戻ります．

```
日本電気㈱■
R〔かな〕
```

このモードから抜けるためには任意の時点で ESC キーを押してください．直前のモードに戻ります．

このモードからJIS16進入力，辞書の切り換え，単語登録，情報表示，辞書の先読みあるいは学習機能の設定モードに入ることはできません．

辞書の先読み機能，学習機能（ f·10 キー）

f·10 キーを押すと，辞書の先読みあるいは学習機能の設定を行うモードとなります．

この機能は，連文節変換と単文節変換で異なりますので，それぞれの場合について説明します．

注意：

・このモードから抜けるためには任意の時点で ESC キーを押してください．直前のモードに戻ります．

・このモードから，1バイトコード入力，JIS16進入力，辞書の切り替え，単語登録，単語削除，情報表示の各モードに入ることはできません．

① 連文節変換の場合

連文節変換では，[f·10] キーを押すと次の画面が表示され，ガイドライン上で学習機能，辞書の先読み機能の設定を行うことができます．

学習機能を設定すると，その後は語句を変換するときに最後に選択された文字がその読みの候補群の先頭となるように候補の順序が入れ替わります．

辞書の先読み機能を設定すると，読みを1文字入力するごとに辞書を読んでデータを記憶しておくので，[XFER] キーによる変換の時間が短縮されます．

```
■
〔設定〕 学習 有 無　　先読み 有 無
```

ガイドラインのキーシフト状態を示す位置には，“設定”と表示されます．

“学習”，“先読み”の変更したい方に [TAB] キーで白地反転のカーソルを合わせ，[→]，[←] キーで“有”，“無”どちらかにカーソルを合わせてください．

リターンキーを押すと設定が終了し，直前のモードに戻ります．

② 単文節変換の場合

単文節変換では，[f·10] キーを押すと次の画面が表示され，ガイドライン上で学習機能の設定を行うことができます．

学習機能を設定すると，その後は語句を交換するときに最後に選択された文字がその読みの候補群の先頭となるように候補の順序が入れかわります．

```
■
〔学習〕 有 無
```

ガイドラインのキーシフト状態を示す位置には，“学習”と表示されます．

現在学習機能が設定されているかどうかはカーソル位置で示されます．

学習機能を設定するには，[←] キーまたは [→] キーでカーソルを指定したい方に動かします．そこでリターンキーを押すとその状態に設定されて直前のモードに戻ります．

情報表示（ [HELP] キー）

[HELP] キーを押すと，つぎの画面が表示され，現在の状態を示します．

表示される情報は，連文節変換では辞書ファイル名，学習機能の有無と辞書の先読み機能の有無で，単文節変換では辞書ファイル名，学習機能の有無です．

(1) 連文節変換での表示例

```
■
〔情報〕 辞書 NECDIC.SYS 学習 有　　先読み 有
```

(2) 単文節変換での表示例

```
■
〔情報〕　辞書　NECDIC.SYS 学習　有
```

どちらもガイドラインのキーシフト状態を示す位置には，“情報”と表示されます．

このモードから抜けるためには，任意の時点で [ESC] キーを押して下さい．直前のモードに戻ります．

このモードから1バイトコード入力，JIS16進入力，辞書の切り替え，辞書の先読みあるいは学習機能設定の各モードに入ることはできません．

部首選択（[CTRL]＋[f･1]）

[CTRL]＋[f･1]を押すと以下の画面が表示され部首選択モードに入ります．

```
■
　〔かな〕　部首
```

直前のモードによる（〔かな〕）

このモードでは入力された読みを部首として扱い変換します．

漢字変換の方法は文節変換の場合と同じです．

例

“いと”を変換します．

① “いと”と入力してください．

```
いと■
　R〔かな〕　部首
```

② [XFER] キーで変換します．

```
綾
　R〔かな〕　部首
```

③ [XFER] キー，[↓] キー，[↑] キーで希望の候補が表示されるまで変換し，リターンキーを押すか次の読みを入力します．これで部首による漢字変換の終了です．[CTRL]＋[f･1] キーを押すと部首選択から抜けます．

緑■

R〔かな〕

途中でこのモードから抜けたい時は，候補が表示されていない時にCTRL ＋ f･1 を押してください．部首選択から抜けます．

部首選択に使える読みは次のとおりです．

画	部首	読み	画	部首	読み	画	部首	読み
一画	一	いち	三画	士	さむらい	四画	方	ほう
	丿	の		夕	ゆう		日	にち
二画	亠	なべ ふた		大	だい		月*	つき
	人	にん ひと		女	おんな		木	き
	儿	ひとあし る		子	こ		欠	けつ あくび
	刂*	りっとう		宀	う		歹	いちた
	刀*	かたな		寸	すん		殳	るまた
	八	はち		小(⺌)	しょう つ		毛	け
	匚,囗,冂	かまえ		尸	しゃく		气	きがまえ
	冖	わ		山	やま		水*	みず すい
	冫	ん に		己	おのれ		爪	つめ
	几	つくえ		巾	はば		片	かた
	凵	うけばこ		广	ま		牛(牜)	うし
	力	か ちから		廴	えん		犬*	いぬ
	勹	つつみ く		弋	しき		礻*	ね
	十	じゅう		弓	ゆみ		王(玉)	おう たま
	卩	ふし せつ		彳	ぎょう		戈	ほこ
	厂	がん		(左)阝(阜)	こざと	五画	瓜	うり
	又	また ぬ		(右)阝(邑)	おおざと		示*	しめす
三画	口	くち ろ		辶(⻌)	しん		衤*	ころも
	氵*	さんし		艹	くさ		田*	た
	犭*	けもの	四画	心*	こころ		疒	やまい
	忄*	りっしん		火*	ひ		癶	はつ
	扌(手)	て		灬*	よつてん		白	しろ
	土	つち ど		攵	のぶん		皮	かわ

画	部　首	読　み	画	部　首	読　み	画	部　首	読　み
	皿	さら	六	血	ち		革	かく
	目	め		衣	きぬ		韋	なめし
五	矢	や	画	襾(西)	にし	九	音	おと
	石	いし		臣	おみ		頁	おおがい ページ
	禾	のぎ		見	みる けん		風	かぜ
	穴	あな		言	ごん		食	しょく
画	立	たつ りつ	七	谷	たに	画	首	くび
	四	よん		豆	まめ		香	かおり こう
	瓦	かわら		豕	いのこ		面	めん
	糸	いと		貝	かい		馬	うま
	缶	かん		赤	あか	十	骨	ほね
	竹	たけ		走	はしる そう		髟	かみ
六	羊	ひつじ		足(⻊)	あし		鬥	とう
	羽	はね		身	み	画	鬼	おに
	老	おい ろう		車	くるま		高	たかい
	耒	すき		辛	からい	十	鳥	とり
	耳	みみ	画	酉	さけ ひよみ	一	魚	うお
	聿	ふで		釆	のごめ	画	鹿	しか
	肉*	にく		豸	むじな		麥(麦)	ばく むぎ
	米	こめ		角	つの	十二画	黑	くろ
	臼	うす		金	かね	十四画	鼻	はな
画	舌	した	八	門	もん	十五画	齒	は
	舟	ふね		隹	ふるとり	十六画	龜	かめ
	虍	とら	画	雨	あめ			
	虫	むし		非	あらず			

（注）＊のついたものは，同一部首で表現が２つあるものを示します．

（忄―心，礻―示，衤―衣，刀―刂，犭―犬，氵―水，火―灬，月―肉）

読み	記号
.か	‘ ’ “ ” ( ) 〔 〕 [ ] { } 〈 〉 《 》 「 」 『 』 【 】 〃 〃
.が	+ − ± × ÷ = ≠ < > ≦ ≧ ∞ ∴ ♂ ♀ ≒ ≡ ∫ ∮ Σ √ ⊥ ∠ ∟ ⊿ ∵ ∩ ∪
.き	♯ & ＊ @ § ※ 〒 〃 仝 々 、 。 , . ・ : ; ? ! ゛ ゜ … ‥ ○ ─ ─ - ／ ＼ ～ → ← ↑ ↓ ＝ ヽ ヾ ゝ ゞ ☆ ★ ○ ● ◎ ◇ ◆ □ ■ △ ▲ ▽ ▼ 〆 ´ ｀ ¨ ＾ ￣ ＿ ‖ ｜
.ぎ	Α Β Γ Δ Ε Ζ Η Θ Ι Κ Λ Μ Ν Ξ Ο Π Ρ Σ Τ Υ Φ Χ Ψ Ω α β γ δ ε ζ η θ ι κ λ μ ν ξ ο π ρ σ τ υ φ χ ψ ω
.け	─ ━ │ ┃ ┄ ┅ ┆ ┇ ┈ ┊ ┉ ┋ ┌ ┍ ┎ ┏ ┐ ┑ ┒ ┓ └ ┕ ┖ ┗ ┘ ┙ ┚ ┛ ├ ┝ ┞ ┟ ┠ ┡ ┢ ┣ ┤ ┥ ┦ ┧ ┨ ┩ ┪ ┫ ┬ ┭ ┮ ┯ ┰ ┱ ┲ ┳ ┴ ┵ ┶ ┷ ┸ ┹ ┺ ┻ ┼ ┽ ┾ ┿ ╀ ╁ ╂ ╃ ╄ ╅ ╆ ╇ ╈ ╉ ╊ ╋
.た	° ′ ″ ℃ ¥ $ ¢ £ % ㍉ ㌔ ㌢ ㍍ ㌘ ㌧ ㌃ ㌶ ㍑ ㍗ ㌍ ㌦ ㌣ ㌫ ㍊ ㌻ mm cm km mg kg cc ㎡
.と	№ ㏍ ℡ ㊤ ㊥ ㊦ ㊧ ㊨ ㈱ ㈲ ㈹ ㍾ ㍽ ㍼
.は	─ ━ │ ┃ ┄ ┅ ┆ ┇ ┈ ┊ ┉ ┋ ┌ ┍ ┎ ┏ ┐ ┑ ┒ ┓ └ ┕ ┖ ┗ ┘ ┙ ┚ ┛ ├ ┝ ┞ ┟ ┠ ┡ ┢ ┣ ┤ ┥ ┦ ┧ ┨ ┩ ┪ ┫ ┬ ┭ ┮ ┯ ┰ ┱ ┲ ┳ ┴ ┵ ┶ ┷ ┸ ┹ ┺ ┻ ┼ ┽ ┾ ┿ ╀ ╁ ╂ ╃ ╄ ╅ ╆ ╇ ╈ ╉ ╊ ╋
.ま	① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨ ⑩ ⑪ ⑫ ⑬ ⑭ ⑮ ⑯ ⑰ ⑱ ⑲ ⑳
.ろ	Ⅰ Ⅱ Ⅲ Ⅳ Ⅴ Ⅵ Ⅶ Ⅷ Ⅸ Ⅹ
.ろしあ	А Б В Г Д Е Ё Ж З И Й К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ъ Ы Ь Э Ю Я а б в г д е ё ж з и й к л м н о п р с т у ф х ц ч ш щ ъ ы ь э ю я
きごう	〒 ※ § ○ ◎ ☆ ゝ ゞ 々 ヽ ヾ + − × ÷ 〆 ○ ‥ ゛ ゜ ´ ｀ ¨ ＾ ￣ ＿ 〃 仝 ─ ─ - ／ ＼ ～ ‖ ｜ … ‘ ’ “ ” 〔 〕 { } 〈 〉 《 》 「 」 『 』 【 】 ± = ≠ < > ≦ ≧ ∞ ∴ ♂ ♀ ° ′ ″ ℃ ¢ £ % ♯ & ＊ ★ ● ◇ ◆ □ ■ △ ▲ ▽ ▼ → ← ↑ ↓ ゛ ゛

（注）．印はピリオドです．テンキーにあるピリオドも使用できます．

5.1.5　キーの使用

日本語入力モードの間，次の各キーは特殊な働きをします．

(1)　モードの切り換えに使用するキー

下記の各キーは，日本語入力モード中，モードや入力文字の切り換えに使用します．
日本語入力モードから抜けるまで通常の働きはしません．

- [カナ]
  カナ文字による入力/半角（またはローマ字）による入力の切り換えに使用します．
- [f･1]
  間接入力モード/直接入力モードの切り換えに使用します．
  辞書がない場合は働きません．
- [f･2]
  ローマ字による入力/英数字による入力の切り換えに使用します．
  カナロック中は働きません．
- [f･3]
  カタカナ入力/ひらがな入力の切り換えに使用します．
  半角の場合は働きません．
- [f･4]
  全角文字/半角文字の切り換えに使用します．
  全角ひらがなの場合，半角に切り換えると半角ｶﾀｶﾅになります．
- [f･5]
  直接入力となりJIS16進コードで1文字入力可能となります．
  読みの入力されている時および候補表示中は働きません．
- [f･6]
  漢字（2バイトコード）/1バイトコードの切り換えに使用します．
- [f･7]
  辞書ファイルの切り換えに使用します．
- [f･8]
  画面上の漢字を辞書へ登録することが可能になります．
- [f･9]
  画面上の漢字を辞書から削除することが可能になります．
- [f･10]
  ガイドラインによって学習機能の設定が行えます．
- [CTRL] + [f･1]
  部首選択モードに入ります．
- [CTRL] + [f･2] ～ [CTRL] + [f･5]
  これらのキーは押しても無視されます．
- [CTRL] + [f･6]
  ガイドラインの表示/未表示を切り換えます．
  表示選択，情報表示，辞書の切り換え，単語登録，単語削除，辞書の先読みあるいは学習機能の

設定，間接入力中はガイドライン未表示にはできません．

・ CTRL ＋ f·7 〜 CTRL ＋ f·10
これらのキーは押しても無視されます．

・ SHIFT ＋ XFER
表示選択モードになります．

・ HELP
辞書と辞書の先読みあるいは学習機能の現在の情報を表示します．

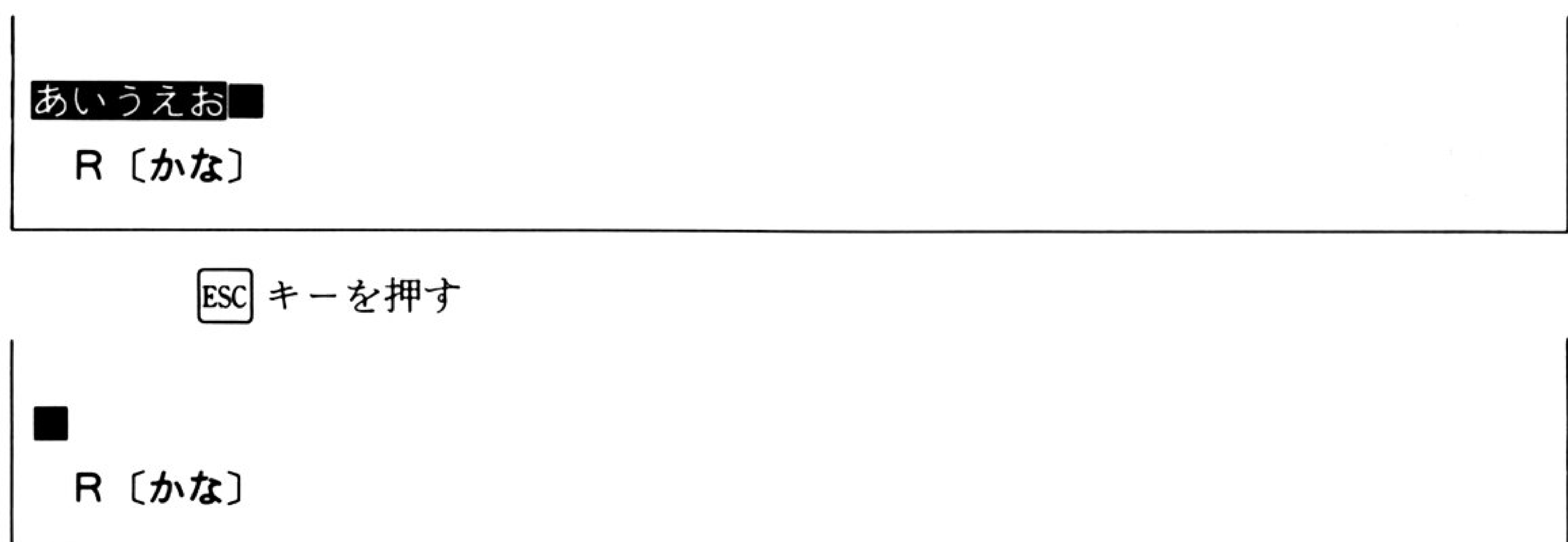
・ ESC
辞書の切り換え，単語登録，単語削除，学習機能の設定，表示選択，部首選択，JIS16進入力，情報表示の時はそのモードから抜けます．

(2) 読みの入力中に使用するキー
下記の各キーは読みが何も入力されていない状態で入力されると通常の働きをします．

・ BS ， ←
カーソル（間接入力中には擬似カーソル）を１文字分左へ移動します．
カーソル（または擬似カーソル）が読みの先頭にあるときは，それ以上左へ移動しません（無視されます）．

・ →
カーソル（または擬似カーソル）を１文字分右へ移動します．
カーソル（または擬似カーソル）が読みの最後の文字の右側にある時はそれ以上移動しません（無視されます）．

・ ESC
読みをすべて消去します．
例
直接入力



間接入力
ガイドラインの読みをすべて消去し，擬似カーソルを読みの先頭に位置づけます．

```
■
  R【かな】　あいうえお■
```

ESC キーを押す

```
■
  R【かな】■
```

・INS

挿入モードになります．

挿入モードではカーソル（または擬似カーソル）のある位置に新しい文字を挿入することができます．

例　あいうえおかきくけこ

INS キーを押す
“お”の入力

あいうえおかきくけこ

文字を挿入したあとカーソル移動キーで擬似カーソルを動かすか，XFER キー等で変換/無変換操作を行うと，挿入モードは解除されます．

・DEL

カーソル（または擬似カーソル）の左側の文字を1文字削除し，カーソル（または擬似カーソル）以降の文字を1文字分左にシフトします．

カーソル（または擬似カーソル）が読みの先頭にあるときはカーソル（または擬似カーソル）の位置にある文字を削除します．

例　**あ**あいうえお

DEL キーを押す

ああいうお

あいうえお

DEL キーを押す

いうえお

・TAB

カーソル（または擬似カーソル）を読みの最後の文字の右側に移動します．

例　あいうえお

TAB キーを押す

あいうえお■

・HOME CLR

カーソル（または擬似カーソル）以降の読みをすべて削除します．

例

```
■
【かな】あいうえおかきくけこ                〈
```

HOME CLR を押す

```
■
【かな】あいうえお■                        〈
```

・SHIFT ＋ HOME CLR

カーソル（または擬似カーソル）を読みの先頭に移動します.

```
あいうえお■
R〔かな〕
```

SHIFT ＋ HOME CLR を押す

```
あいうえお
R〔かな〕
```

(3) 漢字変換時に使用するキー

XFER キーを押して漢字の候補が画面に表示されているとき，下記の各キーは次の働きをします．また，漢字候補が表示されていないときには(2)で示した働き，または通常の働きをします．

・ESC

表示されている漢字候補を取り消して，読みの入力に戻ります．

例（単文節変換）

```
■
R【かな】  漢字を                        〈
```

ESC キーを押す

```
■
R【かな】  かんじを                      〈
```

例（連文節変換）

```
今日私は医者に
R〔かな〕
```

ESC キーを押す

```
きょう私は医者に
R〔かな〕
```

もう一度 ESC キーを押す

```
きょうわたしはいしゃに■

R〔かな〕
```

・→

表示選択のとき，擬似カーソルを次（右側）の漢字候補に重ねます．擬似カーソルが右端の候補に重なっているときには，左端の候補に擬似カーソルを重ねます．

例

```
幹事を
R〔かな〕            1幹事を　2漢字を…
```

→ キーを押す

```
漢字を
R〔かな〕            1幹事を　2漢字を…
```

・←

表示選択のとき，擬似カーソルを前（左側）の漢字候補に重ねます．擬似カーソルが左端の候補に重なっているときには，右側の候補に擬似カーソルを重ねます．

例

```
漢字を
R〔かな〕            1幹事を　2漢字を
```

← キーを押す

```
幹事を
  R〔かな〕          1幹事を  2漢字を
```

・[↑]

1つ前の漢字候補に戻します．最初の候補のときにはそのままです．

表示選択のときは1つ前の候補群に戻します．最初の候補群のときにはそのままです．

例

```
漢字を
  R〔かな〕
```

[↑] キーを押す

```
幹事を
  R〔かな〕
```

・[↓]， [XFER]

次の漢字候補を表示します．最後の候補のときには，読みの表示に戻ります．

表示選択のときは，次の候補群を表示します．最後の候補群のときには，そのままです．

例

```
幹事を
  R〔かな〕
```

[↓] または [XFER] キーを押す

```
漢字を
  R〔かな〕
```

・[NFER]

無変換になります．

読みの表示中，候補の表示中にかかわらず，読みが選択されます．

例

```
漢字を
  R〔かな〕
```

[NFER] キーを押す

かんじを R〔かな〕

・スペースキー（連文節変換の場合のみ）

表示されている全文節を一度に確定させます．

この場合はスペース（空白）は入力されません．

例

私は今日医者へ行きます R〔かな〕

スペースキーを押す

私は今日医者へ行きます■ R〔かな〕

5.2　JISコード入力方式

このモードでは，入力したい日本語を対応するJISコード（16進表記4桁の数字）で入力していきます．

PRINT“日本語”という文字を入力し，実行してみます．

(1)　PRINT”を入力します．

PRINT“■

(2)　CTRL キーを押しながら XFER キーを押します．

JISコード入力のモードになります．

(3)　467Cを入力します．

画面上の“の次に日が表示されます．

PRINT“日■

(4)　4B5Cを入力します．

画面上の日の次に本が表示されます．

PRINT“日本■

(5)　386Cを入力します．

画面上の本の次に語が表示されます．

PRINT“日本語■

(6)　CTRL キーを押しながら XFER キーを押します．

日本語JISコード入力のモードから通常の入力モードにもどります．

(7)　”を入力し最後にリターンキー（↵）を押します．

PRINT“日本語”↵

(8) 次の行に**日本語**と表示されます.

「ハードウエアマニュアル」に日本語コードの一覧表が掲載されています.
各日本語に対応するJISコードはこの表から引いてください.

5.3 辞書ファイルの保守

文節変換入力用辞書ファイル“BUNSET.SU”には約5万語の語句が標準で登録されています.
この辞書には約3%の空き領域が用意されており,この領域を利用して,利用者独自の語句を登録することができます.ただし,“読み”ごとに割当てられている空領域のサイズが異なりますので御注意下さい.
文節変換入力用辞書ファイル“BUNSET.SU”の保守用ユーティリティとして“dicmen.n88”ユーティリティが用意されています.
“dicmen.n88”は次の機能を持っています.

(1) 利用者登録用単語の登録/削除をおこないます.

(2) 辞書ファイルに登録されている単語の一覧表を画面に表示します.

“dicmen.n88”の詳細については,「第15章　システムディスクに格納されているユーティリティプログラム」を参照してください.

第6章
日本語処理

日本語処理機能を使用することにより，日本語文字列を，英数カナの1バイト系文字列と同様に専用高解像度ディスプレイのテキスト画面に表示したり，PC-PR201系プリンタのような漢字出力可能なプリンタに印字したりすることができます．

日本語処理機能を整理すると次のようになります．

(1) 日本語入力

連文節変換入力，単文節変換入力等の入力機能を使用し，キーボードからの日本語入力が可能です．

(2) 文字列定数としての日本語文字列定義

プログラムテキストの中での文字列定数として日本語文字列を定義するもので，英数カナの1バイト系文字列と同様“（クオーテーションマーク）でくくって定義します．

例 「日本語」という文字列をPRINT命令の中で定義する場合

```
print “日本語”
```

(3) 日本語文字列のテキスト画面表示

日本語の文字列をテキスト画面に表示する機能で，日本語1文字は，画面上では2桁の位置を占めます．

日本語文字列としては定数として定義する場合と，文字列変数を引用する場合とがあります．

例 1．「日本語」を画面に表示する．

```
print “日本語” ↵
日本語
```

例 2．「日本語」という日本語文字列を文字列変数a＄に代入し，a＄の内容を表示する．

```
a＄＝“日本語” ↵
print a＄ ↵
日本語
```

(4) 日本語文字列の印刷

日本語文字列を，PC-PR201系の漢字出力可能なプリンタに出力すれば日本語文字を印刷することができます（PC-PR201系プリンタを使用すると明朝体の印字が可能です）．

例 lprint “日本語” ↵

```
日本語
```

(5) グラフィック画面への表示

日本語文字はグラフィック画面へも表示でき，この場合は，1文字単位でPUT＠命令で表示します．

例 「漢」という文字を表示する場合

```
put@(100,100), kanji(&h3441)
```

(6) 日本語文字列の操作

種々の日本語文字列操作用の関数が用意されており，それらの関数を使用することにより，たとえば，日本語を含む文字列の中から一部の文字列を取り出したり，1バイト系の英数カナ文字を日本語に変換したりすることができます．

例 「日本語」という日本語文字列の中から「本」を取り出します．

```
Print chr$(27)+chr$(75)+kmid$("日本語", 3, 1)+chr$(27)+chr$(72) ↵
本
ok
```

(7) 日本語文字列のファイル入出力

日本語文字列も1バイト系英数カナ文字と同じようにファイルに出力することができます．また，当然ながらファイルに出力した日本語を変数に入力することもできます．

例 「日本語」という日本語文字列をディスクファイル（シーケンシャルファイル）に出力し，すぐに読み直します．

```
10 open "DATA" for output as #1
20 print #1, "日本語"
30 close
40 open "DATA" for input as #1
50 input #1, a$
60 print a$
70 close
```

注意 テキスト画面上の日本語文字の文字パターンは16×16となっており画面が25行モードの場合，行間の文字と文字がくっついた形で表示されます．不都合な場合には20行モードで表示してください．

注意 画面に表示する場合も，1文字の日本語が2行にまたがると表示が乱れます．2行にまたがらないよう表示あるいは入力をおこなってください．

また，すでに表示されている上に重ねて表示あるいは入力する場合でも，1文字の日本語を乱すような表示あるいは入力（たとえば，日本語の右半分に英数カナ文字（1バイト文字）を重ねるような表示あるいは入力）をしますと日本語の表示が乱れます．

注意 内部コードが $(0020)_{16}$ から $(007F)_{16}$，$(00B0)_{16}$ から $(00DF)_{16}$ の間にある半角文字を画面表示した場合には，英数カナ1文字の大きさで表示されます．

6.1 日本語文字列の内部形式

(1) 日本語文字は英数カナ2文字と対応します．

日本語文字は，英数カナ2文字分のメモリ（2バイト）を占有します．

(2) 日本語文字列の前後にはシフトコードが付けられます．

日本語文字は，英数カナの日本語文字列の中で英数カナと混在して扱うこともできますが，英

数カナの部分と区別するため日本語文字の始まる前と英数カナに戻る前にそれぞれ漢字イン（KI：1B4B$_H$），漢字アウト（KO：1B48$_H$）が付けられます．また，日本語文字列の内容がすべて日本語の場合でも，最初と最後にKI, KOがそれぞれ付けられます．

このため，桁数を数えたり，比較をしたりする場合は，注意が必要です．

なお，このような桁数のわずらわしさを解決し，日本語文字列の操作をし易くするため，専用の操作関数が用意されています．

（例）“ABC漢字DE”の内部形式

A	B	C	KI	漢	字	KO	D	E	内部形式
41	42	43	1B 4B	34 41	3B 7A	1B 48	44	45	16進コード

6.2　日本語文字列の操作

日本語の文字列も英数カナの文字列と同様に扱えるようになっていますが，日本語文字は，1文字が英数カナ2桁と対応していることと文字列の前後に漢字イン（KI）/漢字アウト（KO）のシフトコードを付けて英数カナ系の文字と区別していることから，文字列の操作や比較がわずらわしくなる点があります．このため，専用の関数を用意し，日本語文字列を扱い易くしています．

日本語文字列操作関数は，日本語を含んでいる可能性のある文字列（英数カナ系の文字が含まれていてもよく，英数カナ系の文字のみでもよい）の操作のためのもので，次のものがあります．

関数名	機　能
AKCNV$	英数カナ系（1バイト系）の文字を対応する日本語系（2バイト系）の文字に変換します．
JIS$	日本語文字を漢字コードで与えます．
KACNV$	日本語系の文字を対応する英数カナ系の文字に変換します．
KEXT$	パラメータの指定により英数カナ系の文字，あるいは日本語系の文字を抜き出します．
KINSTR	指定した文字列があるか否かを調べ，発見された場合は，開始文字位置を返します．
KLEN	パラメータの指定により，指定された文字列の文字数を与えます．
KMID$	指定した位置から始まる指定文字数の文字列を取り出します．
KNJ$	漢字コード（16進表記）を日本語1文字に変換します．
KTYPE	パラメータで指定した位置の文字のタイプを与えます．

注　意　これらの日本語文字列操作関数はROM BASICモードでは使用できません．

6.3　日本語文字列を扱う場合の注意事項

(1)　文字列定数は最大255桁ですが，日本語を含む場合，日本語の1文字が英数カナ2文字に対応しているため，実際の有効文字数はそれ以下となります．また，KI/KOコードも桁数として数えられます．

すべて日本語からなる文字定数の最大文字数は125文字です．

(2)　日本語文字を含んだ文字列の比較を行う場合，シフトコードも含めての比較となります．

(3)　PRINT USING/LPRINT USINGで日本語文字を書式制御文字列に指定したり，日本語文字を含む文字列を編集したりすることはできません．

(4)　KEY命令で日本語をファンクションキーに登録することはできません．

(5)　TAB関数で日本語を制御することはできません．

(6)　日本語文字を含む文字列をスクリーンファイル（"SCRN："）に出力しても日本語の表示は行われません．

(7)　ランダムファイルに日本語文字列を含んだデータを出力する場合，日本語1文字が英数カナ文字2文字に対応すること，また，KI/KOコードが含まれていること等を考えてフィールド長を決定する必要があります．

(8)　画面上に表示されている日本語の左半分あるいは右半分に英数カナ文字（1バイト文字）を重ねて表示しますと日本語の表示が乱れます．

スクリーンエディタを使用し文字を入力する際も，既に入力し終った日本語の左半分あるいは右半分に英数カナ文字（1バイト文字）を重ねて入力したりしますと，日本語の表示が乱れます．

また，日本語が画面の右端で2行にまたがって表示された場合も同様です（ただし，この場合，実際に入力されるデータに影響はありません）．

このような場合，日本語の左半分あるいは右半分が"白ぬき"の状態で表示されます．

6.4　利用者定義文字

JIS第1水準，第2水準の文字を格納したROMに加えて，利用者が任意に文字パターンを設計し，それを日本語文字として扱うことが出来るように，専用のRAMが用意されています（漢字コード7621～767Eおよび7721～777Eの188個の利用者定義文字が使用できます）．

このRAMに文字パターンを設定する命令としてKPLOAD命令があります．KPLOAD命令の使用方法に関しては「BASIC　リファレンスマニュアル」を参照してください．

システムディスク内に，利用者定義文字格納ファイル（ファイル名は"usfont.n88"）が存在すると，自動的に，そのファイル内の文字パターンが専用のRAMへ格納されます．

利用者定義文字格納ファイルの作成，更新にはそのためのユーティリティプログラム（プログラム名は"mkfont. n88"）が用意されています．詳しくは，「第15章　システムディスクに格納されているユーティリティプログラム」を参照して下さい．

第7章

ディスプレイ画面

N88-BASICはいろいろなディスプレイモードを備えています．特に，グラフィック画面の表示用にテキスト画面と独立した大容量のビデオRAM（VRAM）を備えています．

N88-BASICのディスプレイ画面は次のように分類されます．

(1) テキスト画面の種類

モードの名称	表示文字数（横×縦）	備　　　考
40桁20行モード	40×20	モノクロおよびカラーの指定ができます． 漢字を含む日本語表示ができます．
80桁20行モード	80×20	
40桁25行モード	40×25	
80桁25行モード	80×25	

(2) グラフィック画面の種類

モードの名称	分解能（横×縦）ドット数
白黒モード	640×200
カラーモード	640×200
高分解能白黒モード	640×400
高分解能カラーモード	640×400

ディスプレイ装置とグラフィック画面のモードの関係は次の通りです．

ディスプレイ装置の種類 ＼ グラフィック画面モード	専用高解像度ディスプレイ		標準ディスプレイ	
	カラー	モノクロ	カラー	モノクロ
白黒モード	○	○	○	○
カラーモード	○	○*	○	○*
高分解能白黒モード	○	○	×	×
高分解能カラーモード	○	○*	×	×

○：使用できる　○*：濃淡の表示　×：使用できない

7.1 テキスト画面

キーボードから入力した文字や，プログラムのリストなどを表示する画面を「テキスト画面」と呼びます．テキスト画面は，RAMメモリに設定されたテキスト用VRAMを使用して表示します．グラフィック画面とは独立したものです．テキスト画面だけを表示させることも，グラフィック画面にテキスト画面を重ねて表示させることも可能です．

テキストVRAMは12KBあり，アトリビュート（文字の表示状態を保持する情報です．たとえば，ブリンク（点滅），リバース（文字の白黒反転）等の状態を保持するための領域です）は，テキスト1文字に対して1バイトを使用しています．1文字毎の変化があってもかまいません．また，漢字を含む日本語文字を表示することもできます．

注　意　日本語テキストを表示するためには，専用高解像度ディスプレイが必要です．

7.1.1 テキスト画面のサイズ指定

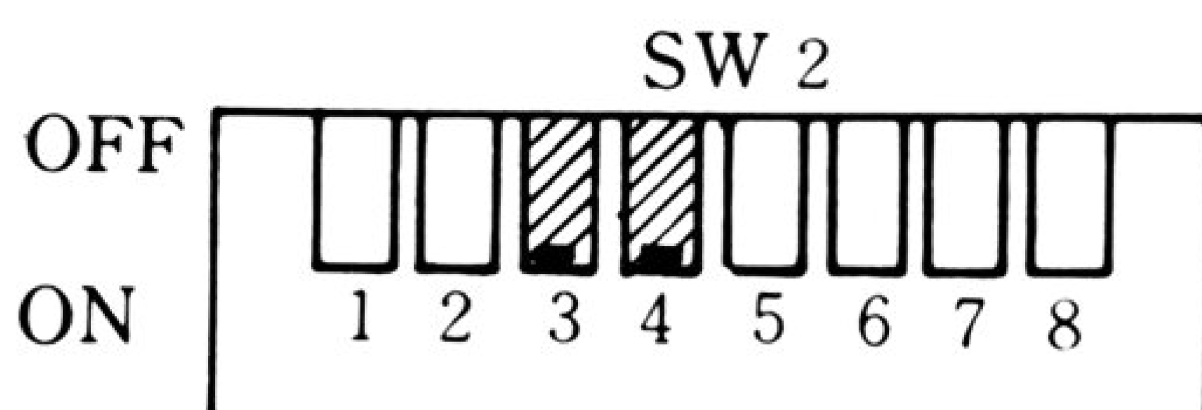
テキスト画面に表示できる文字数は，次の4通りです．

80文字×25行，80文字×20行（80キャラクタモード）

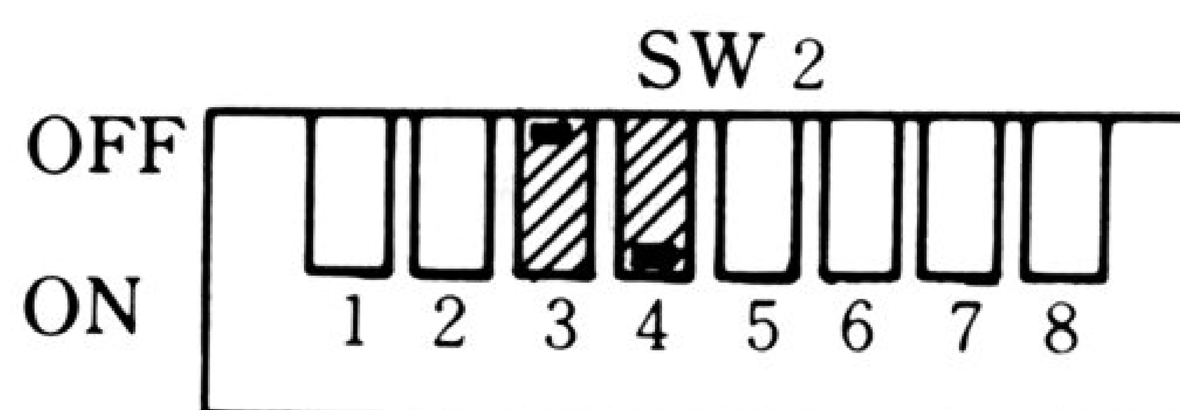
40文字×25行，40文字×20行（40キャラクタモード）

BASIC起動直後の〈文字数/行〉，〈行数/画面〉は，本体前面のディップスイッチSW2で選択できます（出荷時には，80文字×25行の状態にセットされています）．下の図を参照してください．

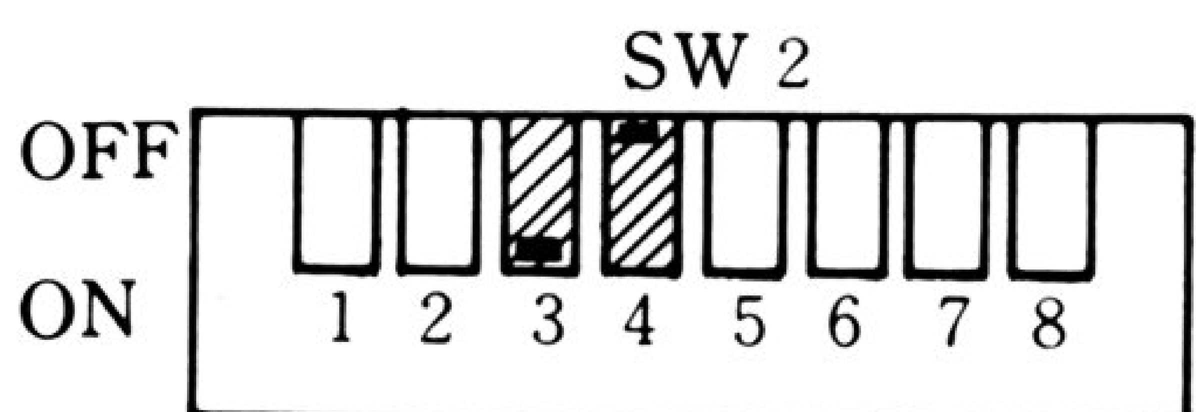
(1) 80文字×25行



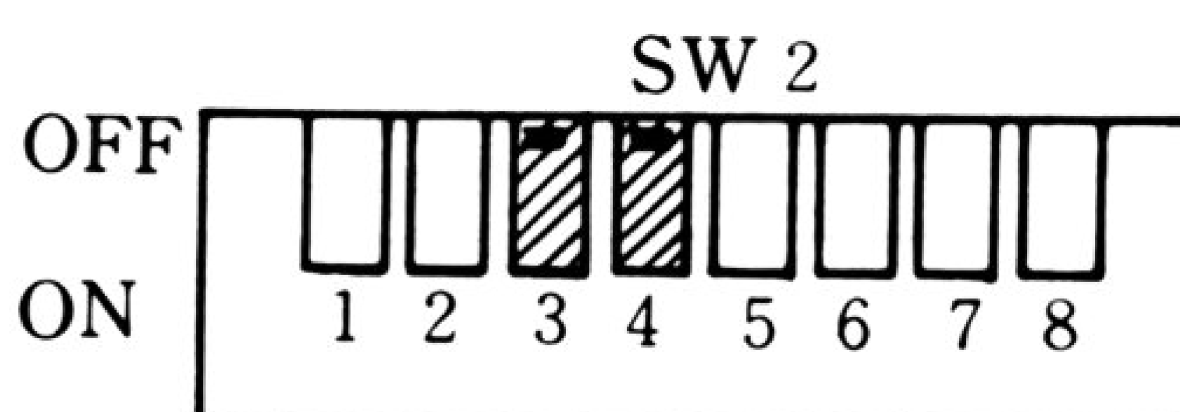
(3) 40文字×25行



(2) 80文字×20行



(4) 40文字×20行



この〈文字数/行〉，〈行数/画面〉はwidth文で変更できます．

次のwidth文による指定があるまで有効です．

width　　〈文字数/行〉，〔〈行数/画面〉〕 ↵

例　40文字×20行の画面を80文字×25行に変更する場合

```
width 80, 25 ↵
```

7.1.2　テキスト画面の条件設定

テキスト画面の表示条件には，次の4つがあります．

スクロールのスタート行
スクロールする行数
ファンクションキーの表示
カラー/モノクロモードの指定

(1)　スクロール画面

ディスプレイの画面は，処理したデータの内容を時々刻々と表示してゆくような，変化する部分と，見出しのように同じ内容を表示しつづける静止した状態の部分とに分けることができます．

画面の変化する動作のことを「スクロールする」とよび，画面の変化する領域のことを「スクロール画面」とよびます．

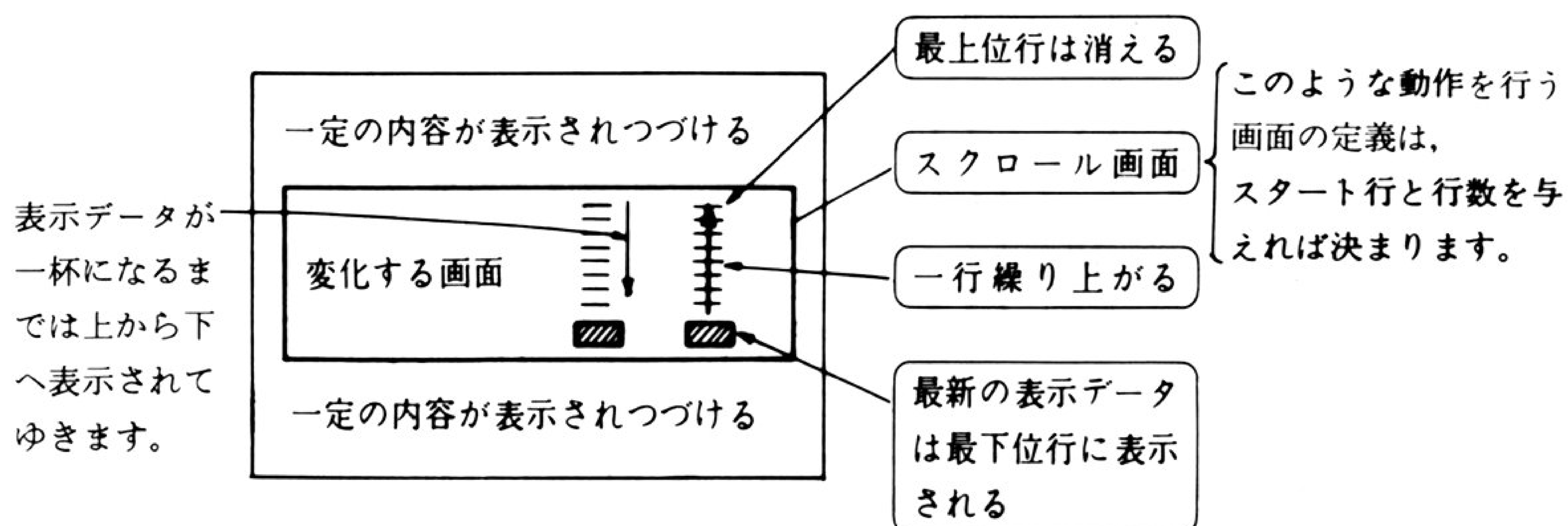
テキスト画面のスクロール方法は次のようになっています．

①　スクロール画面が表示データで一杯になるまでの動作

スクロール画面に，まだ何も表示されていない状態では，スクロール画面の最上位行に最初のデータが表示されます．新しい表示データは，順次，スクロール画面の最下位行に向って表示されてゆきます．最新の表示データが最下位行に表示されると，スクロール画面は表示データで一杯になります．このような状態では，最も先に表示された古い表示データが最上位行に表示されており，最下位行に向って，順次，新しいデータが表示されています．最下位行が最も新しい表示データになっています．

②　スクロール画面が表示データですべて使用されている場合の動作

最も古い表示データが，最上位行から消え，スクロール画面上の表示データが全体として，1行上へ移動します．空白になった最下位行に最新の表示データが表示されます．スクロール画面が表示データで一杯になっている状態ではこの動作が繰り返されます．



(2) ファンクションキーの表示

画面の最下位行には，ファンクションキーを表示することができます．

80文字/行画面では，[f·1] から [f·10] までの10個のキーが表示できます．

40文字/行画面では，[f·1] から [f·5] までの５個のキーが表示できます．それぞれのキーは，上位６文字までの表示です． [SHIFT] キーを押すと， [f·6] から [f·10] までの５個のキーが表示されます．

テキスト画面の条件設定は，console文で行います．

console 〔〈スクロールのスタート行〉〕，〔〈スクロールする行数〉〕，〔〈ファンクションキーの表示〉〕，〔〈カラー/白黒スイッチ〉〕

注 意 第１パラメータのスタート行の指定方法は，画面の行を最上位から０，１，２……と数えます．１画面25行モードのときは，行の最下位は24です．

第１パラメータの値

……０（１行目）

……１（２行目）

……24（25行目）

例 最上位行から10行を残しておきたい場合，第０行から第９行を残すように設定しますから，〈スクロールのスタート行〉は，10，〈スクロールする行数〉は，20行/画面ならば，残り10行ですから，これも10となります．

console　10，10

すると，上半分の10行は，スクロールされません．

0 → a a a

9 → j j j j……

10 → 10 { k k k k … t t t t

もとの状態にもどすには，第０行から，20行数をスクロールするようにセットします．

console　0,20

例 〈ファンクションキーの表示〉は，表示する場合は１，表示しない場合は０を指定しま

す．

表示しない　console ,,0

表示する　console ,,1

〈カラー/モノクロの指定〉は，カラーにする場合は1，モノクロの場合は0を指定します．

モノクロ　console ,,,0

カラー　console ,,,1

例　5行目から，14行目までをスクロールさせ，ファンクションキーは表示せず，カラーの指定を行う場合

console 4, 10, 0, 1

7.1.3　テキスト画面のカラー指定

テキスト画面のカラー指定は，color文で行います．N₈₈-BASICには，3種類のcolor文があります．このうち，テキスト画面のカラー指定を行うのは，次の2つです．

(1)　ファンクションコード/フォアグラウンドカラー/バックグラウンドカラー/ボーダーカラーの指定を行うcolor文．

color　〔〈ファンクションコード〉〕，〔〈バックグラウンドカラー〉〕，〔〈ボーダーカラー〉〕，〔〈フォアグラウンドカラー〉〕

この中で，テキスト画面に関係があるのは，〈ファンクションコード〉だけで，他はグラフィック画面に関係あるものです．ここでは，〈ファンクションコード〉だけを説明し，残りのパラメータは，「第8章　グラフィックス」の中で説明します．

「7.1.2　テキスト画面の条件設定」で説明したconsole文の4番目のパラメータに，〈カラー/白黒スイッチ〉があります．ファンクションコードは，この指定によって異なった意味を持ちます．

カラー指定の場合(console,,,1)

〈ファンクションコード〉	色	濃淡または明暗
0	黒	淡(暗)
1	青	↓
2	赤	↓
3	紫	↓
4	緑	↓
5	水色	↓
6	黄色	↓
7	白	濃(明)

モノクロ指定の場合(console,,,0)

〈ファンクションコード〉	文字の表示モード
0	ノーマル
1	シークレット
2	ブリンク
3	シークレット
4	リバース
5	リバース・シークレット
6	リバース・ブリンク
7	リバース・シークレット

〈ノーマル〉とは，通常の表示モードです．〈リバース〉にすると背景と文字の色が逆転します．〈シークレット〉は，入力された文字を表示せず，文字数分だけカーソルが移動していきます．〈ブリンク〉は文字を点滅させます．

例 テキスト画面の文字を赤にしたい場合．

console ,,,1 ⏎ （カラーモード）

color 2 ⏎ （赤）

この２つの命令により，これ以後入力される文字は，赤で表示されます．

例 暗証番号のように，ディスプレイには表示させたくない場合．

console ,,,0 ⏎ （モノクロモード）

color 1 ⏎ （シークレット）

BASIC起動直後には，モノクロモードとなっていますから，このconsole文は省略できます．

(2) テキスト画面に書かれた文字などに，色や機能を区画で指定するcolor文．

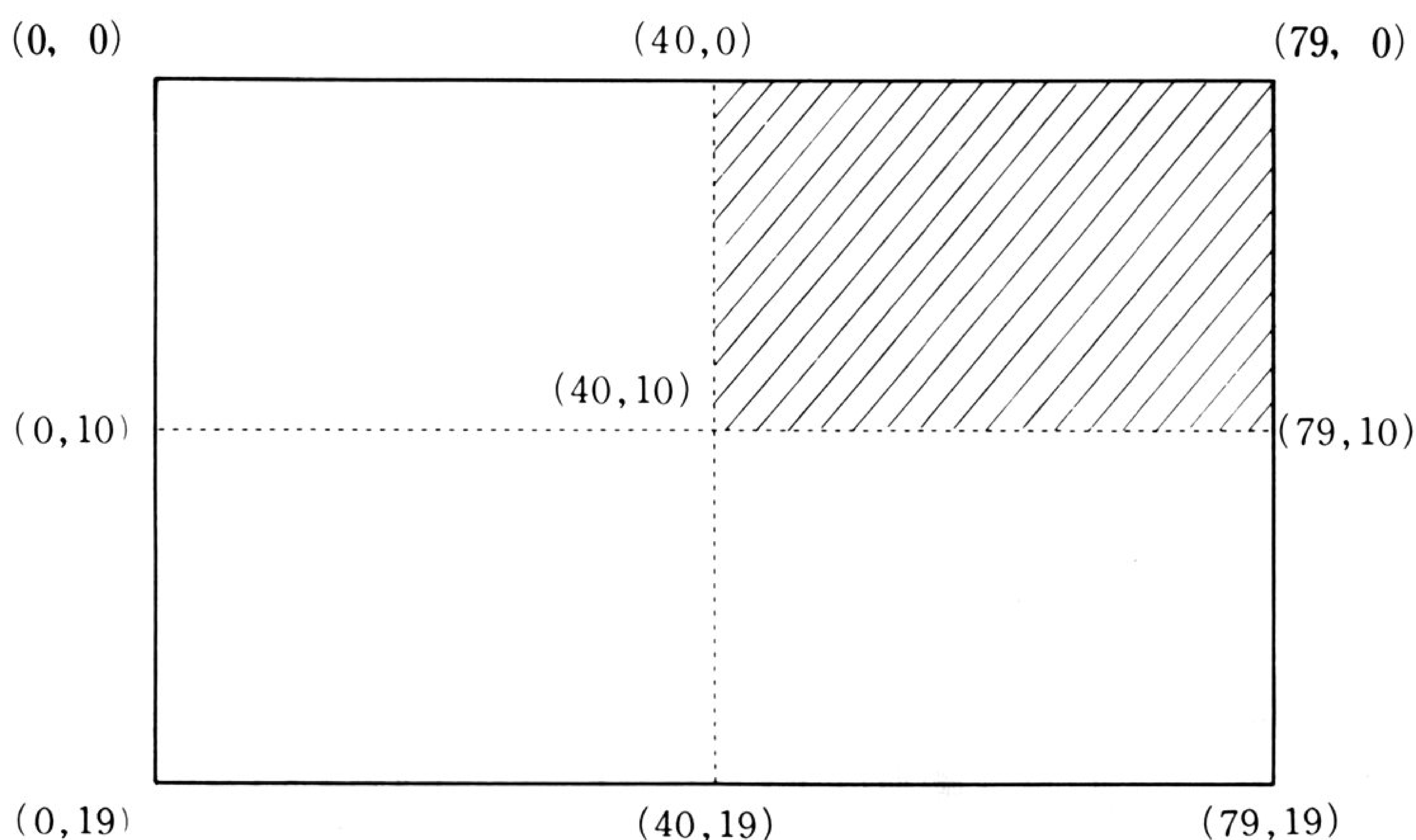
N_{88}-BASICでは，テキスト画面の，ある区画を，〈ファンクションコード〉によって指定することができます．

color@ $(X_1, Y_1)-(X_2, Y_2)$，〈ファンクションコード〉 ⏎

(X_1, Y_1)，(X_2, Y_2) は，キャラクタ座標（文字位置）です．

〈ファンクションコード〉については，(1)で説明したとおりです．キャラクタ座標は，「何文字目で何行目」というように位置を指定するもので，画面の左上スミが，（０，０）となります．

例 80文字×20行モードのとき，画面の右上半分のすでに表示されているテキスト文字の色を青色にする場合．



区画を指定するときは，その区画の左上スミと右下スミを指定します．

console ,,,1 ↵

color @ (40, 0) — (79, 10), 1 ↵

同じ区画に表示されている文字を点滅（ブリンク）させる場合は，次のようになります．

console ,,,0 ↵

color @ (40, 0) — (79, 10), 2 ↵

注意1 座標はキャラクタ座標で指定します．指定できる範囲は，次のようになります．
（0，0）～（〈文字数/行〉－1，〈行数/画面〉－1）．

注意2 このcolor文では，すでに表示されている文字の色を変えることができますが，次にこの区間に表示される文字は，(1)のcolor文で指定された〈ファンクションコード〉によって表示されます．

7.2 グラフィック画面

グラフィック画面のモードには，〈白黒モード〉と〈カラーモード〉があります．

さらに、モノクロまたはカラーの専用高解像度ディスプレイを使用している場合は，640×400ドットというきわめて鮮明な画像が得られる〈高分解能白黒モード〉または〈高分解能カラーモード〉を選択することができます．

7.2.1 画面モード

グラフィック画面のモードは，screen文を使います．

この節では8色モード時の画面の取扱いについて説明します．16色モード時の画面の取扱いについては「7.2.2 (3) 16色モード時の画面数」を参照してください．

screen 〔〈画面モード〉〕〔,〈画面スイッチ〉〕〔,〈アクティブ画面〉〕〔,〈ディスプレイ画面〉〕

・〈画面モード〉

指定できる値	モード	分解能(横×縦)	画面数
0	カラーモード	640×200	4
1	白黒モード	640×200	12
2	高分解能白黒モード	640×400	6
3	高分解能カラーモード	640×400	2

・〈画面スイッチ〉

指定できる値	機　　能
0	グラフィック画面への表示がおこなわれる．
1	
2	現在グラフィック画面に表示されている情報が一時的に消去されます．
3	

・〈アクティブ画面〉

グラフィック命令を実行する画面を指定します．

画面モードによって，取り扱える画面数が異なります．

指定できる値	モ　ー　ド	指定できる値と画面との対応関係
0～3	カラーモード	0：第1画面～3：第4画面
0～11	白黒モード	0：第1画面～11：第12画面
0～5	高分解能白黒モード	0：第1画面～5：第6画面
0，1	高分解能カラーモード	0：第1画面，1：第2画面

例えば，カラーモードでは，第1画面，第2画面，第3画面，第4画面が指定できます．

① 第1画面に書き込む場合は，パラメータ3の値は0です．

② 第4画面に書き込む場合は，パラメータ3の値は3です．

・〈ディスプレイ画面〉

表示する画面を指定します．

パラメータの値と表示する画面は，ビット対応になっています．白黒モードおよび高分解能白黒モードにおいては，3つまでの画面を重ねて表示できる画面の合成機能があります．

たとえば，白黒モードの場合最大12個の画面を扱えますが，これを第1画面から第3画面，第4画面から第6画面，第7画面から第9画面および第10画面から第12画面の4つのグループに分けて，それぞれのグループ内での重ね表示を許すことにしています．全体で32通りあります．

パラメータの値は，5ビットを使い，その中の2ビットは画面グループを選択するビットに使い，他の3ビットは，それぞれの画面グループ内の重ね表示の組を表わすために使います．

具体的には，次の表をみてください．

ビット配置	3，4	2	1	0
扱う画面グループ(第1)	0，0	第3画面	第2画面	第1画面
扱う画面グループ(第2)	1，0	第6画面	第5画面	第4画面
扱う画面グループ(第3)	0，1	第9画面	第8画面	第7画面
扱う画面グループ(第4)	1，1	第12画面	第11画面	第10画面

表の見方　①　ビット3，4が0，0ならば，第1グループの3つの画面が対象になります．
ビット3，4が1，0ならば，第2グループの3つの画面が対象になります．

②　例えば，第1画面と第3画面の重ね表示をする場合は，
$1 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + 0 \cdot 2^3 + 0 \cdot 2^4 \rightarrow 5$　パラメータ4は5

③　例えば，第5画面と第6画面の重ね表示をする場合は，
$0 \cdot 2^0 + 1 \cdot 2^1 + 1 \cdot 2^2 + 1 \cdot 2^3 + 0 \cdot 2^4 \rightarrow 14$　パラメータ4は14

・パラメータの値と表示する画面，使用できる画面モードとの関係は次のようになります．

ディスプレイページの値	画面モードごとの意味			
	0	1	2	3
0	全画面表示しない	全画面表示しない	全画面表示しない	全画面表示しない
1	画面1のみ表示	画面1のみ表示	画面1のみ表示	画面1のみ表示
2	画面2のみ表示	画面2のみ表示	画面2のみ表示	×
3	×	画面1，2を合成表示	画面1，2を合成表示	×
4	×	画面3のみ表示	画面3のみ表示	×
5	×	画面1，3を合成表示	画面1，3を合成表示	×
6	×	画面2，3を合成表示	画面2，3を合成表示	×
7	×	画面1，2，3を合成表示	画面1，2，3を合成表示	×
8	全画面表示しない	全画面表示しない	全画面表示しない	全画面表示しない
9	×	画面4のみ表示	×	×
10	×	画面5のみ表示	×	×
11	×	画面4，5を合成表示	×	×
12	×	画面6のみ表示	×	×
13	×	画面4，6を合成表示	×	×
14	×	画面5，6を合成表示	×	×
15	×	画面4，5，6を合成表示	×	×
16	全画面表示しない	全画面表示しない	全画面表示しない	全画面表示しない
17	画面3のみ表示	画面7のみ表示	画面4のみ表示	画面2のみ表示
18	画面4のみ表示	画面8のみ表示	画面5のみ表示	×
19	×	画面7，8を合成表示	画面4，5を合成表示	×
20	×	画面9のみ表示	画面6のみ表示	×
21	×	画面7，9を合成表示	画面4，6を合成表示	×
22	×	画面8，9を合成表示	画面5，6を合成表示	×
23	×	画面7，8，9を合成表示	画面4，5，6を合成表示	×
24	全画面表示しない	全画面表示しない	全画面表示しない	全画面表示しない
25	×	画面10のみ表示	××	×

26	×	画面11のみ表示	×	×
27	×	画面10, 11を合成表示	×	×
28	×	画面12のみ表示	×	×
29	×	画面10, 12を合成表示	×	×
30	×	画面11, 12を合成表示	×	×
31	×	画面10, 11, 12を合成表示	×	×

×印：指定不可

(1) 高分解能カラーモード

専用高解像度カラーディスプレイを接続した場合は，640×400ドットの鮮明なカラー表示を行う高分解能カラーモードが選択できます．

高分解能カラーモードの画面は2枚です（合成はできません）．

カラー表示については，通常，システムが規定した8色の表示が可能ですが，アナログRGB対応ディスプレイが接続されている場合はさらに4096色中の任意の8色あるいは16色を選んで表示することができます．

高分解能カラーモードを指定する場合は，次の命令を実行してください．

screen 3, 0, 〈パラメータ3〉, 〈パラメータ4〉 ↵

専用高解像度モノクロディスプレイを接続した場合はカラーに相当する濃淡の表示ができます．

(2) 高分解能白黒モード

専用高解像度モノクロディスプレイを接続した場合640×400ドットの高分解能白黒モードを選択できます．

高分解能白黒モードは，6枚の画面が使用でき，書き込み画面と表示画面とを区別した操作ができます．また，画面の合成もできます．

高分解能白黒モードを指定する場合は，次の命令を実行してください．

screen 2, 0, 〈パラメータ3〉, 〈パラメータ4〉 ↵

(3) 白黒モード

640×200ドットの白黒モードの画面は12枚あります．書き込み画面と表示画面を区別した操作ができます．また，第1画面から第3画面，第4画面から第6画面，第7画面から第9画面および第10画面から第12画面の4組について，それぞれの組の中で，2画面，または3画面の合成もできます．

白黒モードを指定する場合は，次の命令を実行してください．

screen 1, 0, 〈パラメータ3〉, 〈パラメータ4〉 ↵

例 第1画面でグラフィック命令を実行させながら，その画面をディスプレイに表示させる場合

screen 1, 0, 0, 1 ↵

例 第6画面でグラフィック命令を実行させながら，第4画面と第5画面を表示させる場合

screen 1, 0, 5, 11 ↵

この場合は，実行されているグラフィック命令の結果は，ディスプレイには現われません．実行終了後に次の命令を実行すると，第6画面で実行された結果も合わせて表示されます．

screen　1, 0, 5, 15 ⏎

(4)　カラーモード

640×200ドットのカラー表示を行うカラーモードでは4枚の画面が使用できます（合成はできません）.

カラー表示については，高分解能カラーモードと同様に通常システムが規定した8色の表示が可能ですが，アナログRGB対応ディスプレイが接続されている場合はさらに4096色中の任意の8色あるいは16色を選んで表示することができます.

カラーモードを指定する場合は，次の命令を実行してください.

screen　0, 0,〈パラメータ3〉,〈パラメータ4〉 ⏎

モノクロディスプレイでカラーモードを使用すると，カラーに相当する濃淡の表示が可能です.

7.2.2　基本グラフィックモードと拡張グラフィックモード

システムのグラフィック機能モードには基本グラフィックモードと拡張グラフィックモードの2つがあり，いずれのモードでシステムが起動されるかにより，グラフィック機能（特にカラー指定の方法と機能）に違いがあります.

	基本グラフィックモード	拡張グラフィックモード
カラー表示	システムが規定した8色が使用可能	(1)システムが規定した8色が使用できるモード (2)4096色中の任意の8色が使用できるモード (3)4096色中の任意の16色が使用できるモード 上記3種のモード（これをパレットモードと呼びます）のいずれかを選択することができます.
高速描画機能	高速描画のためのハードウエアは使用しません.	高速描画のためのハードウエアを描画時に適用します.

(1)　基本グラフィックモードと拡張グラフィックモードの選択

この2つのモードは起動時に一意に決まるものです．システムを立ち上げた後，切り換えながら使用していくものではありません.

	BASICのモード	基グラフィックモード/拡張グラフィックモード	
1	ROM BASICモード	無条件に基本グラフィックモードとなります.	
2	DISK BASICモード	拡張グラフィックスイッチがONの場合	拡張グラフィックモードとなります.
		拡張グラフィックスイッチがOFFの場合	基本グラフィックモードとなります.

備　考　**拡張グラフィックスイッチはDIPスイッチSW1の8番スイッチ（SW1の右端のスイッチです）を指します.**

このスイッチは出荷時にはONの状態（すなわち，拡張グラフィックモードの状態）になっています.

注　意　**拡張グラフィックモードでシステムが起動された場合，拡張グラフィック手続き（22KB）が利用者RAM（DISK CODE部）に追加ロードされます．その分，利用者メモリ領域が圧迫されますので御注意ください.**

(2) カラー表示

(a) 基本グラフィックモードにおけるカラー

次の8色が使用できます.

黒，明るい青，明るい赤，明るい紫，明るい緑，明るい水色，明るい黄色，白

(b) 拡張グラフィックモードにおけるカラー

・8色中・8色モード

基本グラフィックモードの8色と同じ8色が使用できます.

・4096色中・8色モード（アナログRGB対応ディスプレイが接続されている場合に有効です）

4096色中の任意の8色が使用できます.

オレンジ色，少し暗い緑，灰色等の中間色の表示が可能なモードです.

・4096色中・16色モード（アナログRGB対応ディスプレイが接続されている場合有効です）

4096色中の任意の16色が使用できます.

4096色中・8色モードと同様，中間色の表示が可能なモードです.

注　意　4096色中・8色および4096色中・16色モードはアナログRGB対応ディスプレイが接続されている場合のみ有効です.

アナログRGB対応ディスプレイ以外のディスプレイが接続されている場合にこれらのモードで中間色を表示しようとしてもエラーにはなりませんが，指定通りの色は表示されませんので御注意ください.

カラー指定に関する詳細については「第8章　8.2　カラー指定」を参照してください.

(3) 16色モード時の画面数

4096色中・16色モード時の画面数は次のとおりです.

モ　　ー　　ド	分 解 能	画 面 数
カラーモード	640×200	4
白黒モード	640×200	16
高分解能白黒モード	640×400	8
高分解能カラーモード	640×400	2

また，screen文の〈アクティブ画面〉と〈ディスプレイ画面〉に指定可能な値は次のとおりです．

・〈アクティブ画面〉

指定できる値	モ　　ー　　ド	指定できる値と画面との対応関係
0～3	カラーモード	0：第1画面～3：第4画面
0～15	白黒モード	0：第1画面～15：第16画面
0～7	高分解白黒モード	0：第1画面～7：第8画面
0，1	高分解カラーモード	0：第1画面，1：第2画面

・〈ディスプレイ画面〉

ディスプレイページの値	画面モードごとの意味			
	0	1	2	3
0	全画面表示しない	全画面表示しない	全画面表示しない	全画面表示しない
1	画面１のみ表示	画面１のみを表示	画面１のみ表示	画面１のみ表示
2	画面２のみ表示	画面２のみ表示	画面２のみ表示	×
3	×	画面１，２を合成表示	画面１，２を合成表示	×
4	×	画面３のみ表示	画面３のみ表示	×
5	×	画面１，３を合成表示	画面１，３を合成表示	×
6	×	画面２，３を合成表示	画面２，３を合成表示	×
7	×	画面１，２，３を合成表示	画面１，２，３を合成表示	×
8	×	画面４のみ表示	画面４のみ表示	×
9	×	画面１，４を合成表示	画面１，４を合成表示	×
10	×	画面２，４を合成表示	画面２，４を合成表示	×
11	×	画面１，２，４を合成表示	画面１，２，４を合成表示	×
12	×	画面３，４を合成表示	画面３，４を合成表示	×
13	×	画面１，３，４を合成表示	画面１，３，４を合成表示	×
14	×	画面２，３，４を合成表示	画面２，３，４を合成表示	×
15	×	画面1,2,3,4を合成表示	画面1,2,3,4を合成表示	×
16	全画面表示しない	全画面表示しない	全画面表示しない	全画面表示しない
17	×	画面５のみ表示	×	×
18	×	画面６のみ表示	×	×
19	×	画面５，６を合成表示	×	×
20	×	画面７のみ表示	×	×
21	×	画面５，７を合成表示	×	×
22	×	画面６，７を合成表示	×	×
23	×	画面５，６，７を合成表示	×	×
24	×	画面８のみ表示	×	×
25	×	画面５，８を合成表示	×	×
26	×	画面６，８を合成表示	×	×
27	×	画面５，６，８を合成表示	×	×
28	×	画面７，８を合成表示	×	×
29	×	画面５，７，８を合成表示	×	×
30	×	画面６，７，８を合成表示	×	×
31	×	画面5,6,7,8を合成表示	×	×
32	全画面表示しない	全画面表示しない	全画面表示しない	全画面表示しない
33	画面３のみ表示	画面９のみ表示	画面５のみ表示	画面２のみ表示
34	画面４のみ表示	画面10のみ表示	画面６のみ表示	×
35	×	画面９，10を合成表示	画面５，６を合成表示	×
36	×	画面11のみ表示	画面７のみ表示	×

ディスプレイ画面の値	画面モードごとの意味			
	0	1	2	3
37	×	画面９，11を合成表示	画面５，７を合成表示	×
38	×	画面10，11を合成表示	画面６，７を合成表示	×
39	×	画面９，10，11を合成表示	画面５，６，７を合成表示	×
40	×	画面12のみ表示	画面８のみ表示	×
41	×	画面９，12を合成表示	画面５，８を合成表示	×
42	×	画面10，12を合成表示	画面６，８を合成表示	×
43	×	画面９，10，12を合成表示	画面５，６，８を合成表示	×
44	×	画面11，12を合成表示	画面７，８を合成表示	×
45	×	画面９，11，12を合成表示	画面５，７，８を合成表示	×
46	×	画面10，11，12を合成表示	画面６，７，８を合成表示	×
47	×	画面９，10，11，12を合成表示	画面5，6，7，8を合成表示	×
48	全画面表示しない	全画面表示しない	全画面表示しない	全画面表示しない
49	×	画面13のみ表示	×	×
50	×	画面14のみ表示	×	×
51	×	画面13，14を合成表示	×	×
52	×	画面15のみ表示	×	×
53	×	画面13，15を合成表示	×	×
54	×	画面14，15を合成表示	×	×
55	×	画面13，14，15を合成表示	×	×
56	×	画面16のみ表示	×	×
57	×	画面13，16を合成表示	×	×
58	×	画面14，16を合成表示	×	×
59	×	画面13，14，16を合成表示	×	×
60	×	画面15，16を合成表示	×	×
61	×	画面13，14，15，16を合成表示	×	×
62	×	画面14，15，16を合成表示	×	×
63	×	画面13，14，15，16を合成表示	×	×

第8章
グラフィックス

8.1 グラフィック画面の座標系

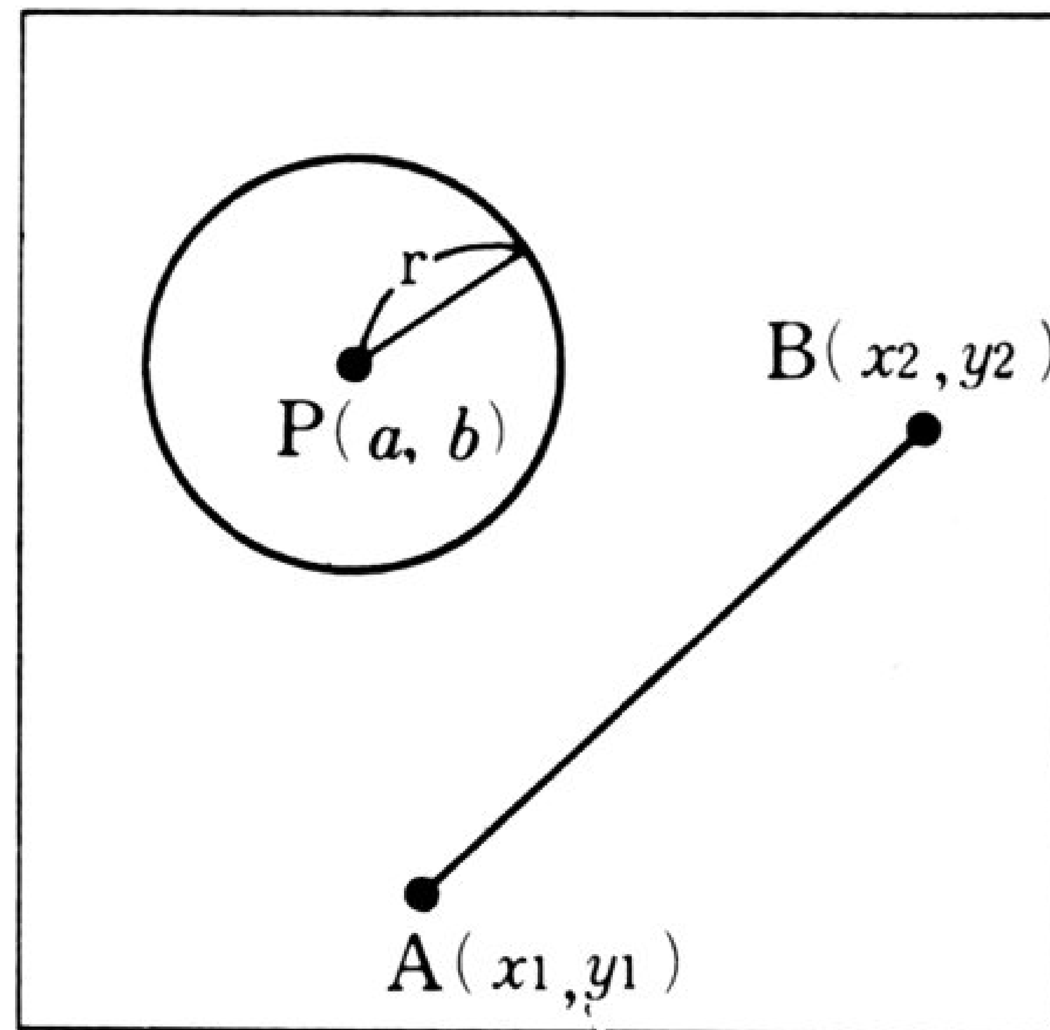
8.1.1 3つの座標系，ウィンドウ，ビューポート

複雑な図形を描いたり，図形の移動，拡大，縮小の操作を容易に行うため3つの座標系が用意されています．

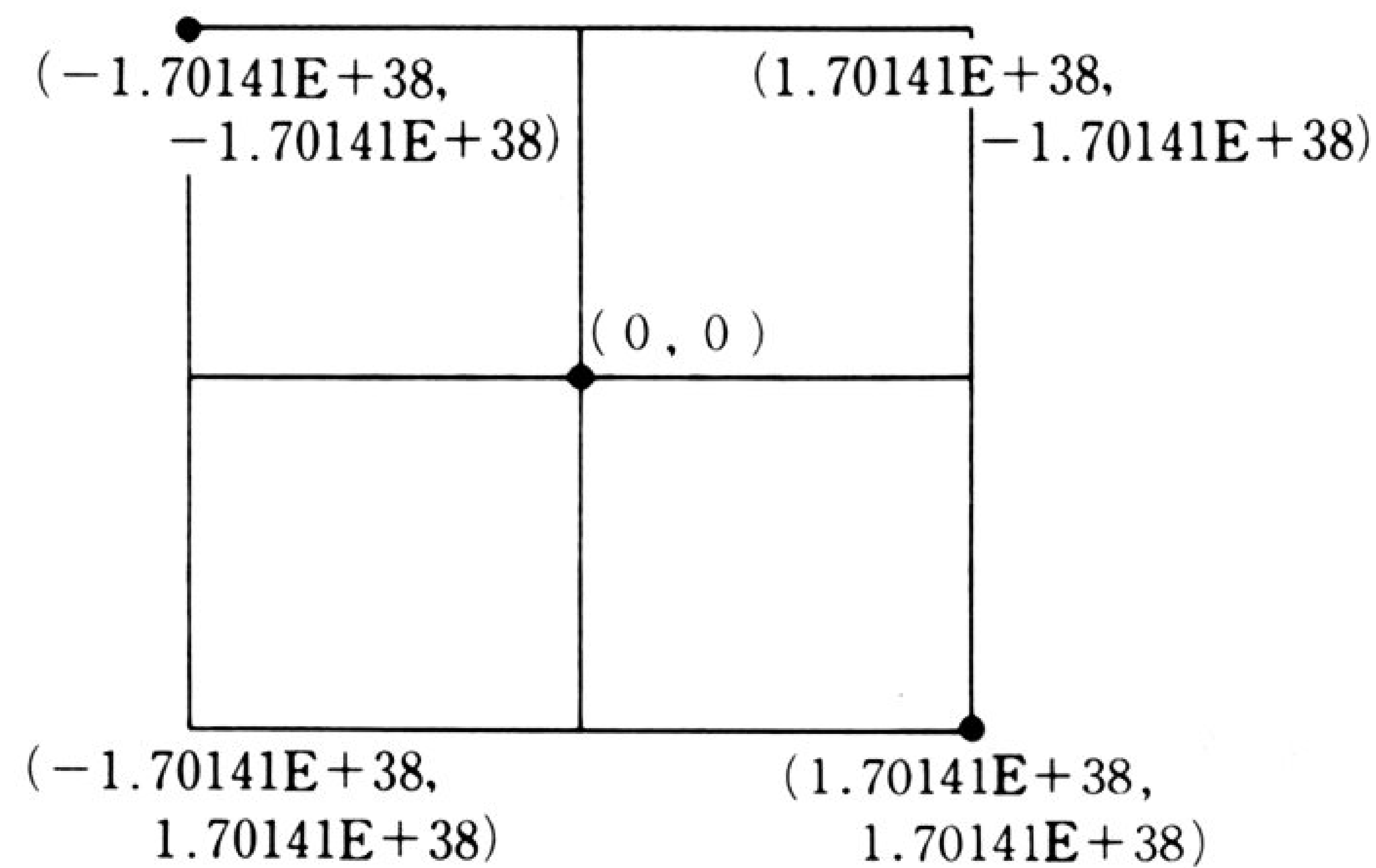
(1) ワールド座標系

・グラフィックスで使用するプログラミング上の座標系を“ワールド座標系”とよびます．
たとえば，点P(a, b)を中心とする半径rの円を描くとか，2点A(x_1, y_1)，B(x_2, y_2)を結ぶ線分ABを描くには，論理的なワールド座標系を使用します．

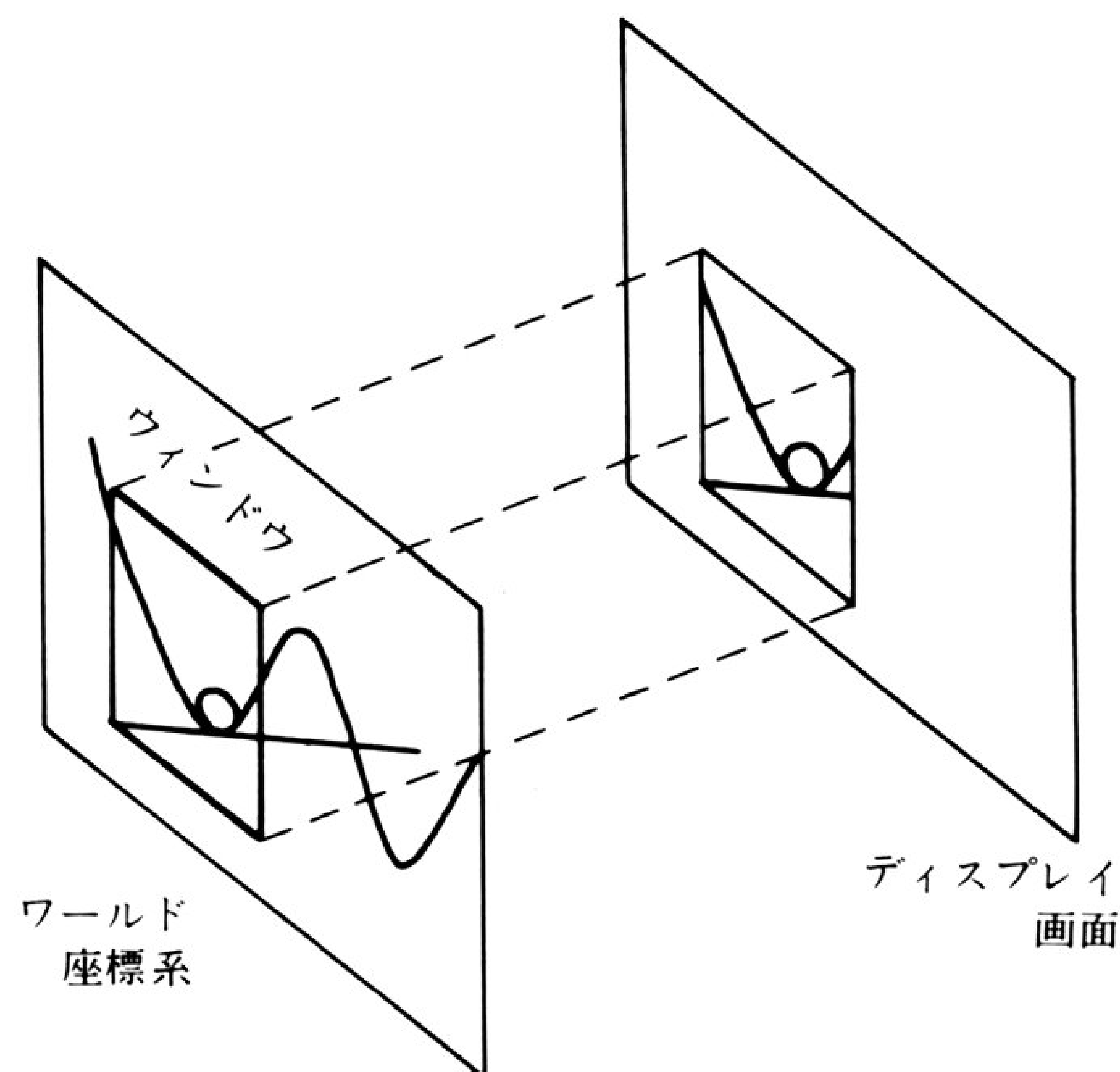
・ワールド座標系は，論理的な座標系で，実数型数値で表現できる大きさの座標をもっています．



CIRCLE文，LINE文を使用します。



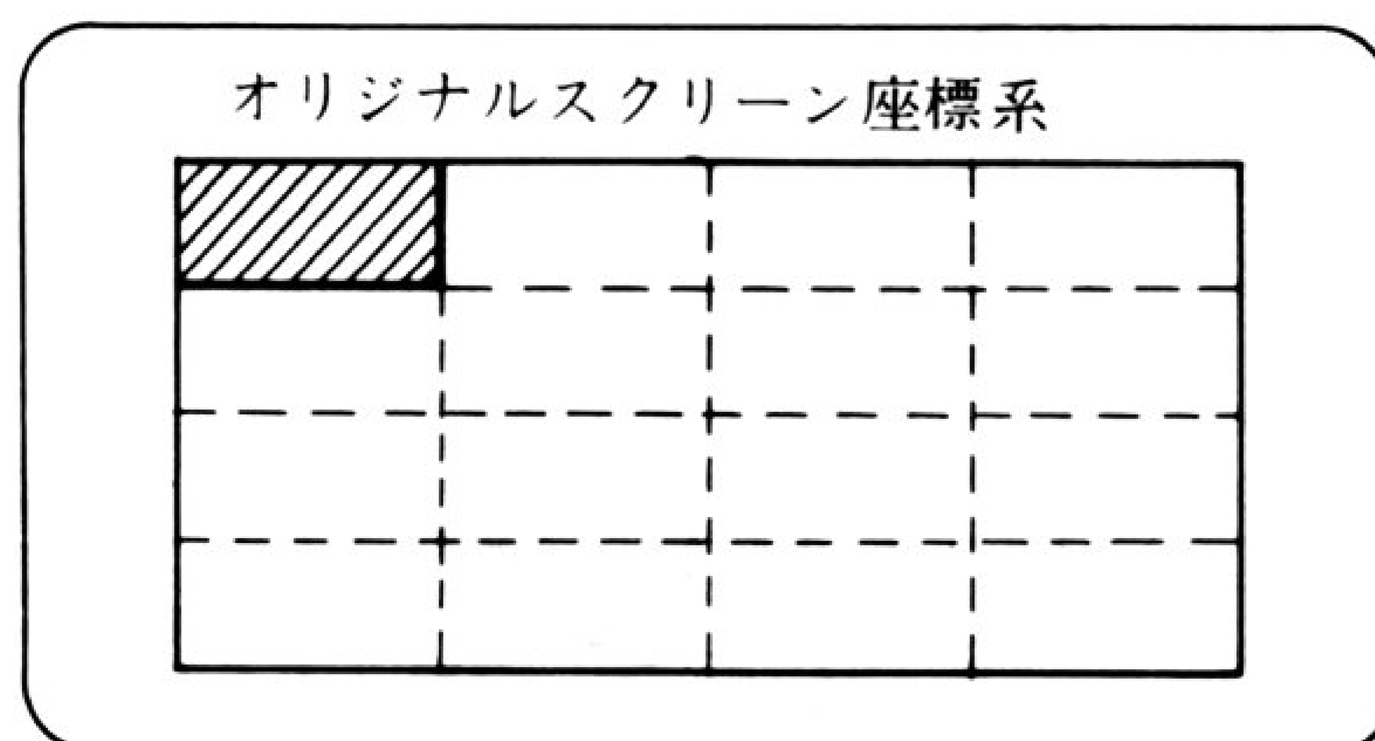
・ワールド座標系で表わしている図形の中から，実際にディスプレイに写し出す領域のことを
“ウィンドウ（のぞきまど）”とよびます．
論理的には複雑な図形を描いているのですが，ウィンドウを通して表示する領域は，その時点で必要な部分だけをとりだします．この部分が画面へ投影する対象になります．
この領域を**window**文で定義します．

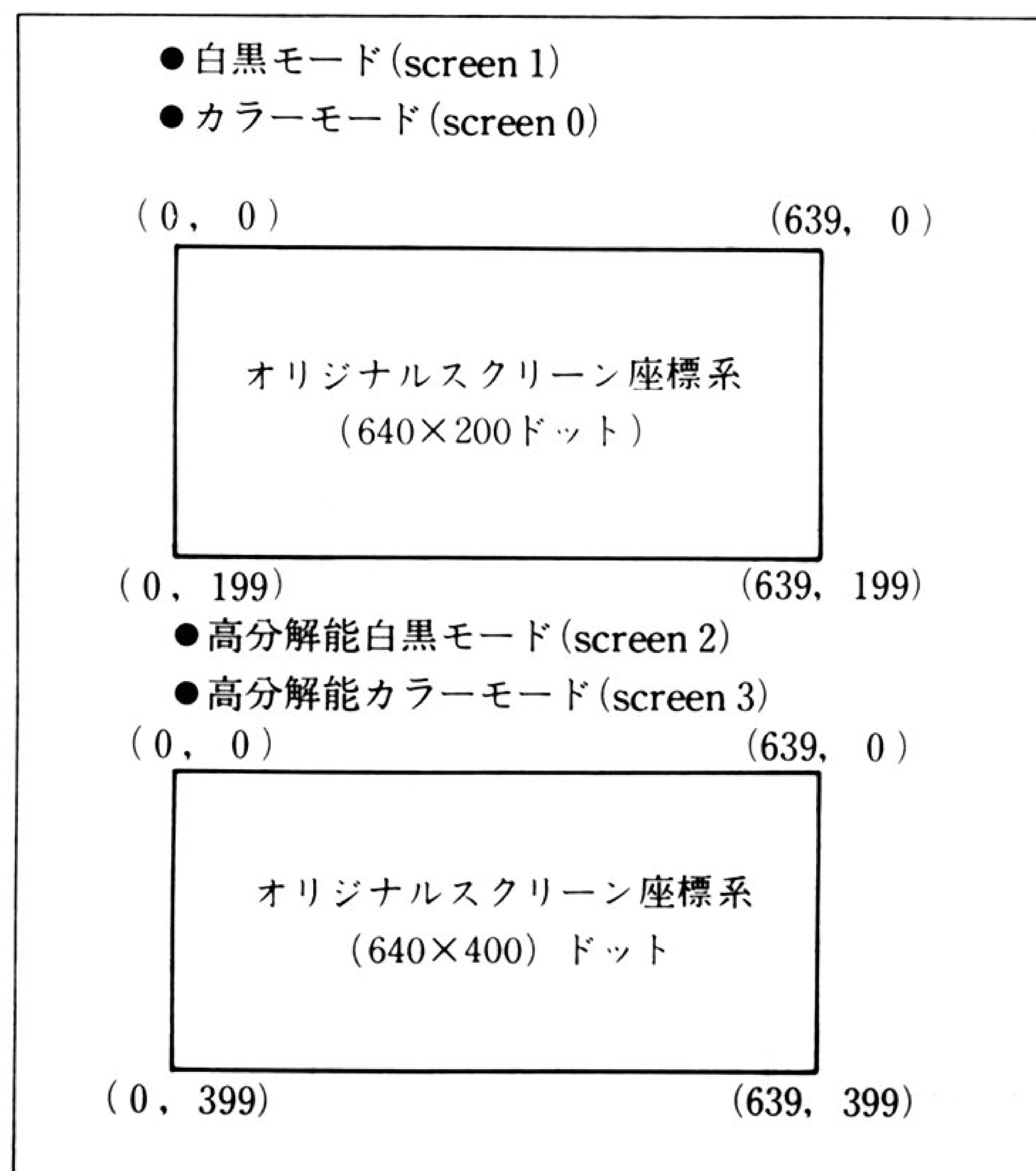


(2) オリジナルスクリーン座標系

実際に，ディスプレイ装置のブラウン管で表示される図形は，ドットで表わされています．このドットを単位としたディスプレイに，直接結びついた座標系を“オリジナルスクリーン座標系”とよびます．最も身近な座標系です．物理座標です．

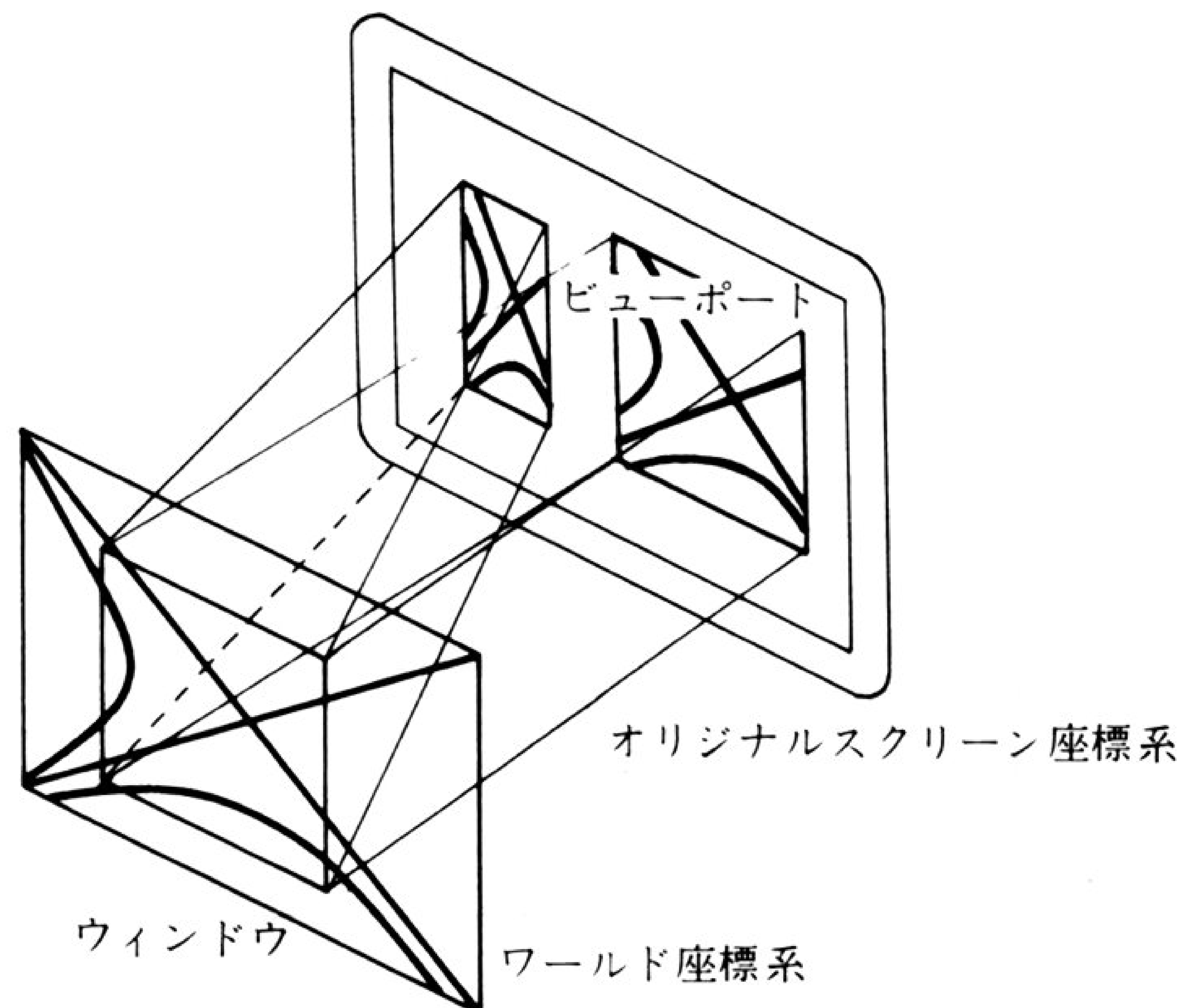
たとえば，次図のように，ディスプレイの左上，1/16の領域に図形を表示するような場合に，このオリジナルスクリーン座標系を使って表わします．





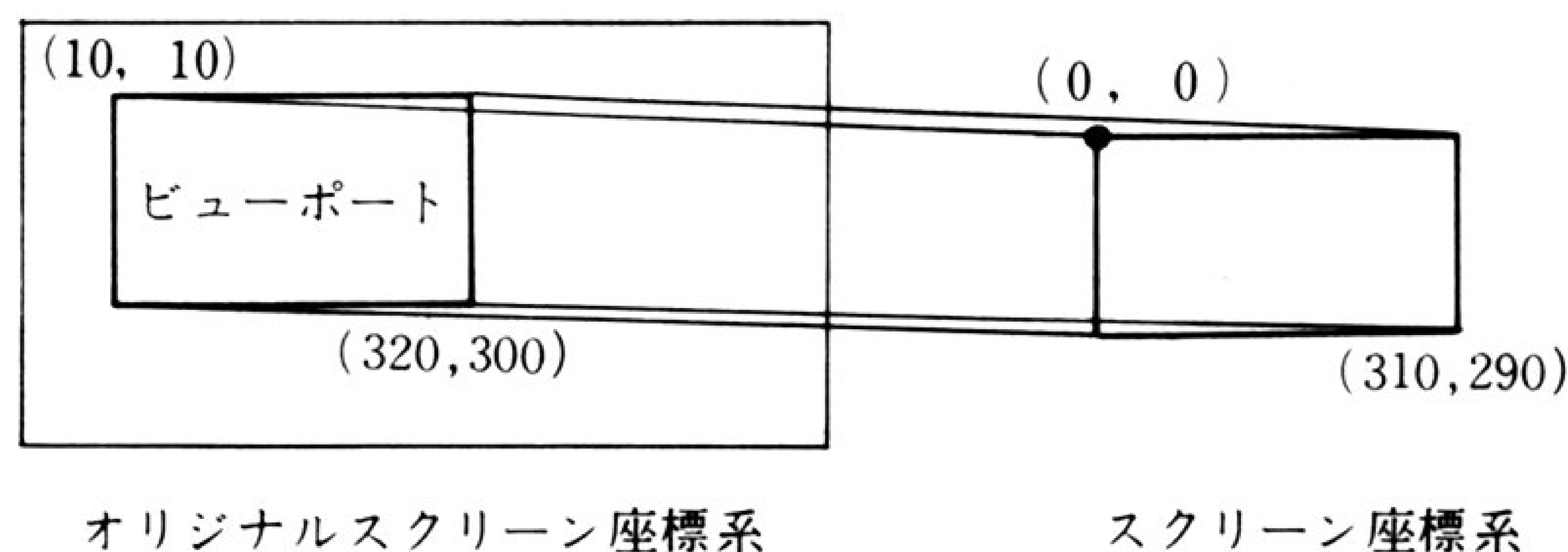
オリジナルスクリーン座標系の大きさは画面モードによって決まります．もちろん実際に表示されるかどうかは，この画面モードとディスプレイ装置の分解能との対応が正しいものでなくてはいけません．

・ワールド座標系の中の図形で，ウィンドウによって，とりだした領域を，オリジナルスクリーン座標系に写し出します．オリジナルスクリーン座標系中で，これから写し出される領域のことを，**“ビューポート”**とよびます．ビューポートを定めるためには，オリジナルスクリーン座標系で表わされる唯一の命令であるview文を使います．
view文の指定で，ビューポートを定めると，ビューポートの長方形の領域が，指定した領域色で塗りつぶされ，境界は指定した境界色で囲いが引かれます．



(3) スクリーン座標系

・オリジナルスクリーン座標系で表わされるディスプレイ画面の中で，これからの操作の対象になる領域がビューポートです．ビューポート以外の画面は以前の状態が持続しています．この活動中の領域を，新しい座標系で見直しますと，これからの操作がしやすくなります．
ビューポートの左上を基点（0，0）として，ドットを単位に座標をとりなおしたのが，スクリーン座標系です．



・スクリーン座標系は，ビューポートの操作をしやすくします．
たとえば，ビューポート内の図形をドットパターンでメモリ上の配列にセーブする場合（get@文を使います），この図形の領域を指定するのに，基点がビューポートの左上にあると，セーブする配列とビューポート内の領域との対応が容易につきます．移動座標よりも，正規化されている方が処理しやすいでしょう．
put@文もスクリーン座標系で表わします．画面の指定したドットの色を知るためのPOINT関数も，この座標系を使います．

(4) 座標系の関係

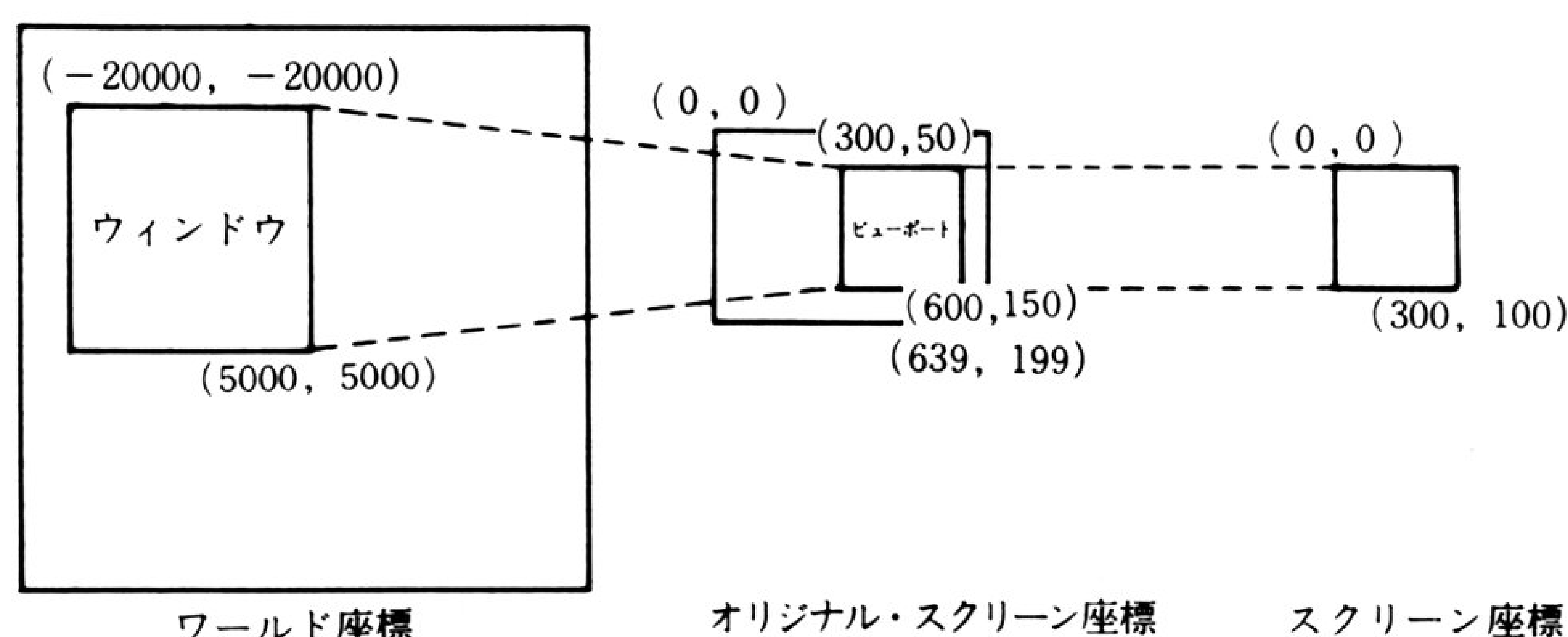
ワールド座標系，オリジナルスクリーン座標系，スクリーン座標系の3つの座標系の関係，ウィンドウとビューポートの関係を，次にまとめておきます．

① ワールド座標系で，複雑な図形をプログラミングするために，LINE文，CIRCLE文，POINT文等を使用します．

複雑な図形の中から，ディスプレイに表示する領域をウィンドウとしてとりだします.

② オリジナルスクリーン座標系で，ディスプレイする画面上の領域をビューポートとして定めます.
ウィンドウ内の図形がビューポートに写しだされます.

③ ビューポート内の左上を基点としたスクリーン座標系で，ビューポート内の図形を見直します.
ディスプレイ上の図形のドットイメージをメモリ上の配列に格納したり，メモリ上の配列に格納されている図形のドットイメージをディスプレイに写しだしたりします.



(5) 各座標系のもとで使用できるグラフィック制御命令

詳しくは「BASICリファレンスマニュアル」を参照してください.

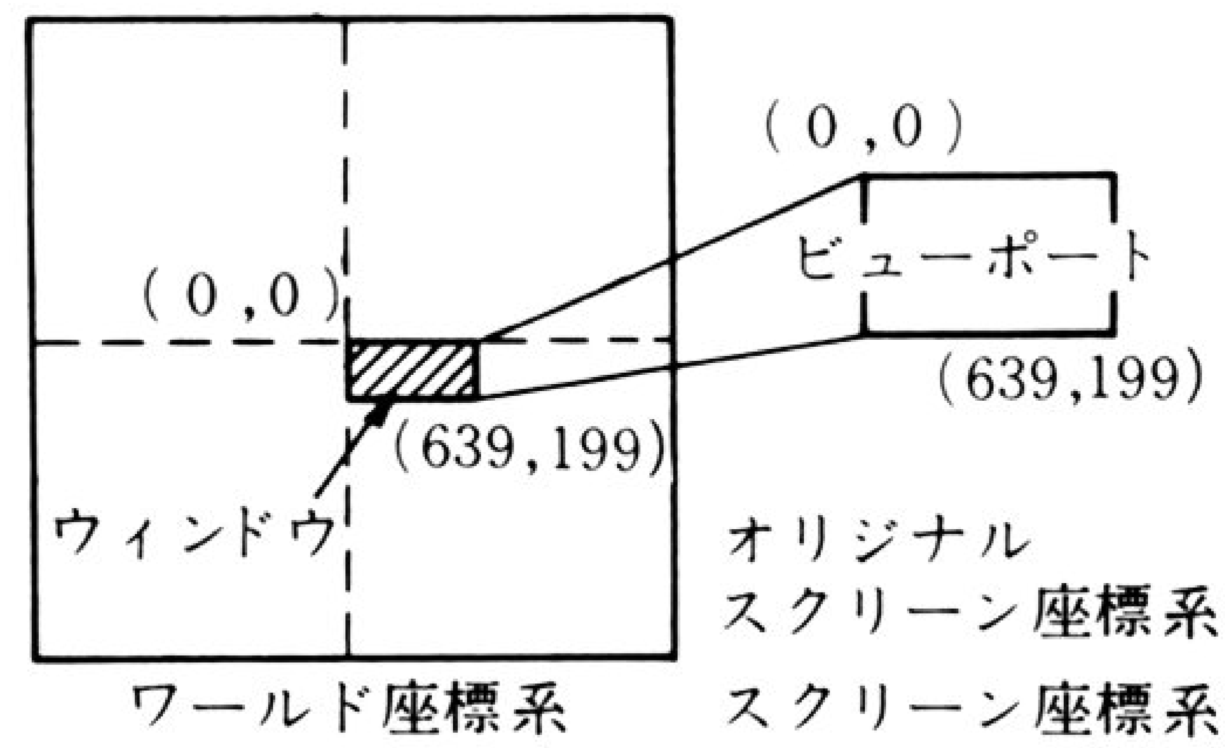
座標系	ワールド座標系	オリジナルスクリーン座標系	スクリーン座標系
使用できる文	CIRCLE DRAW LINE PAINT POINT PRESET PSET WINDOW	VIEW	GET@ POINT（関数） PUT@
座標の表わし方	(Wx_1, Wy_1)	(Sx_1, Sy_1)	(Sx_1, Sy_1)

(6) ウィンドウとビューポートの初期値

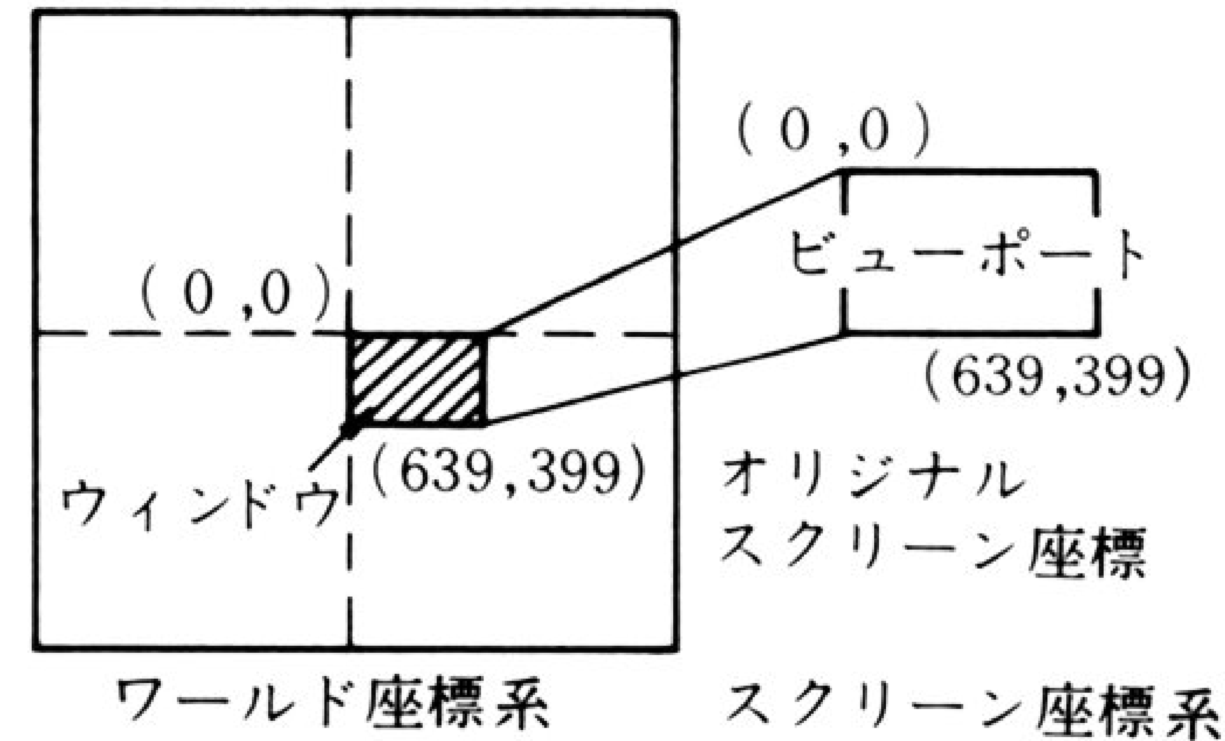
screen文で画面モードを指定したときのウィンドウとビューポートはディスプレイのドット数の大きさに初期化されます.

window文，view文によって変更するまで，この値が使われます.

①カラーモード(screen 0)
白黒モード (screen 1)



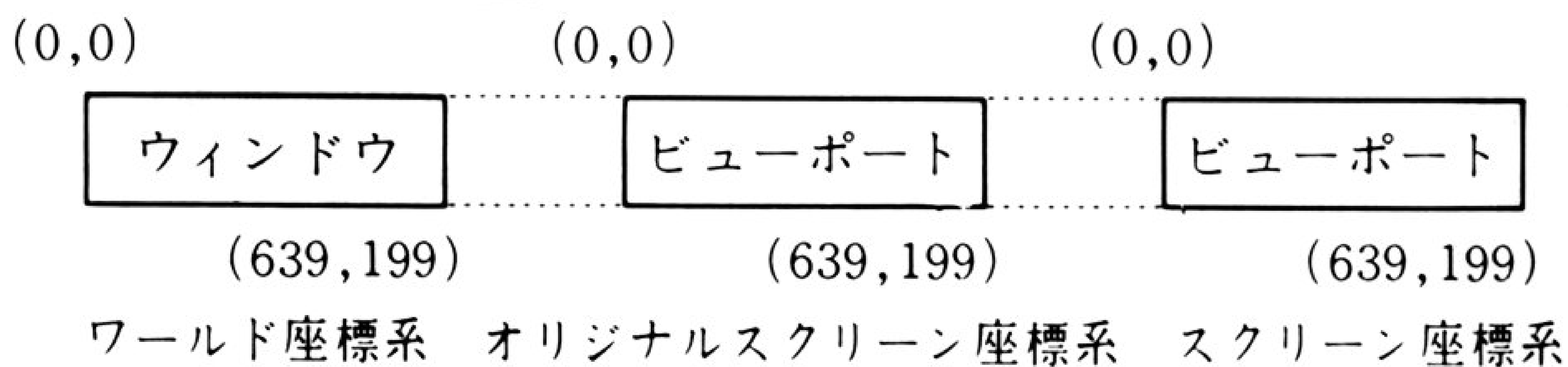
②高分解能白黒モード (screen 2)
高分解能カラーモード(screen 3)



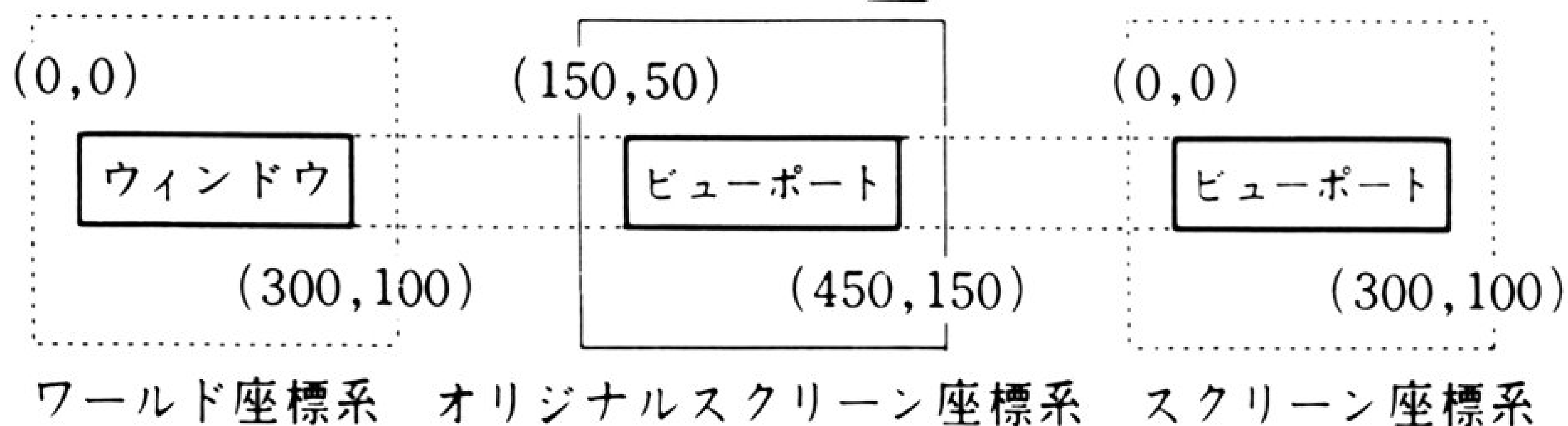
(7) window文でウィンドウを指定しないで，view文でビューポートを指定すると，ウィンドウが変化します．注意してください．

(a) window文で指定しない場合

① screen 0,0 [↵]



② view(150,50)−(450,150),, 7 [↵]

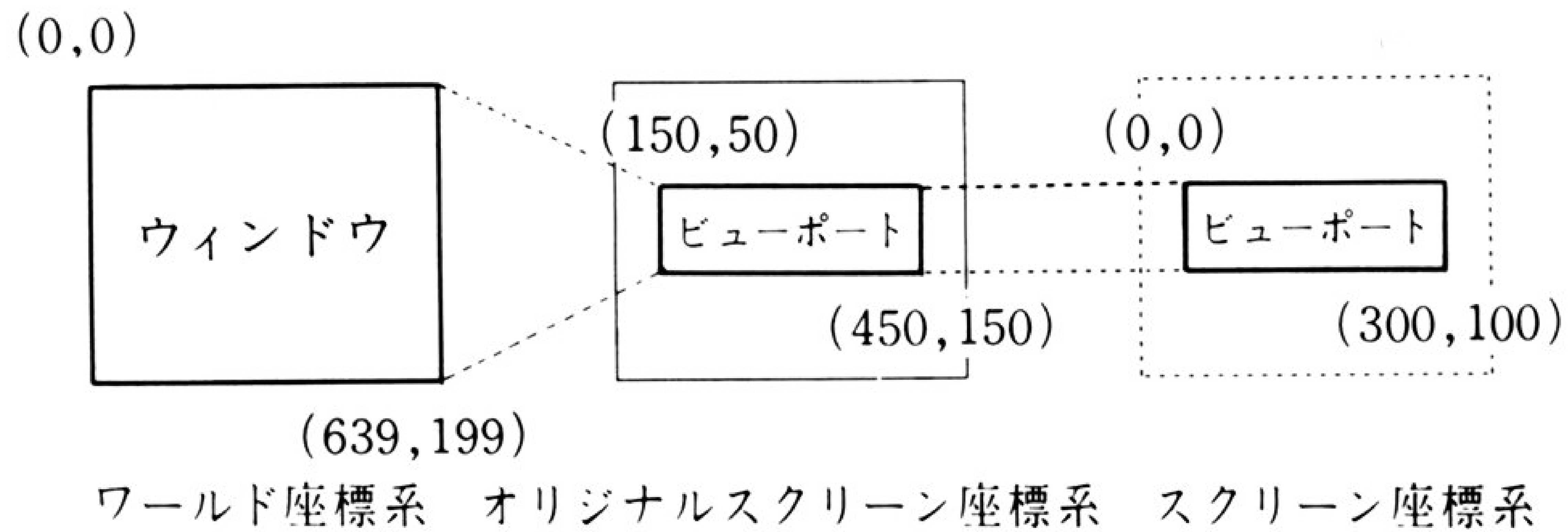


(b) window文を指定した場合

① screen 0,0 [↵]

② window (0,0)−(639,199) [↵]

view (150,50)−(450,150),, 7 [↵]



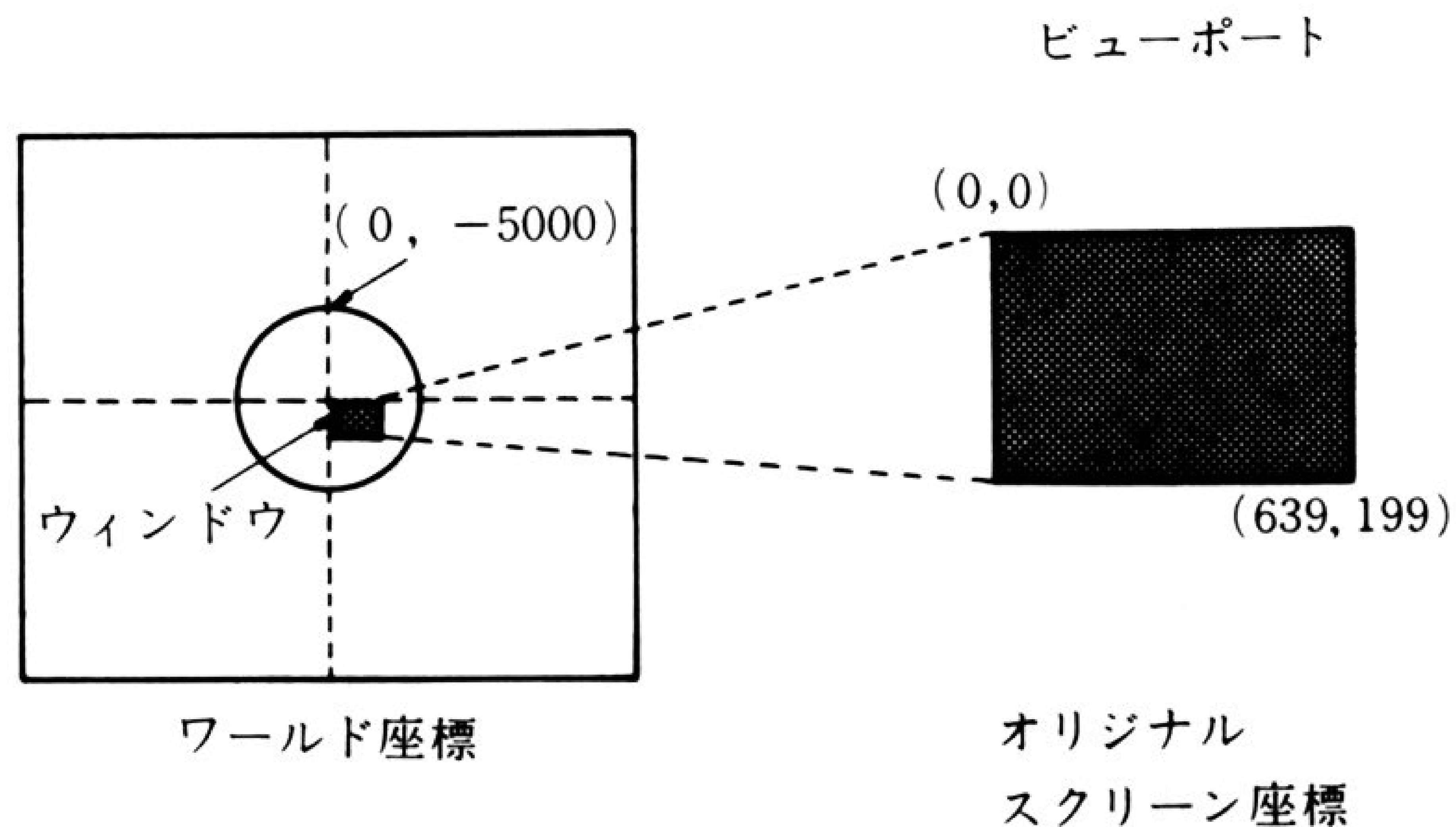
例　次の命令を実行してみます．

```
screen　0，0⏎
circle（0，0），5000，1⏎
```

・画面には何もみえません
・ウィンドウとビューポートは初期値の状態です

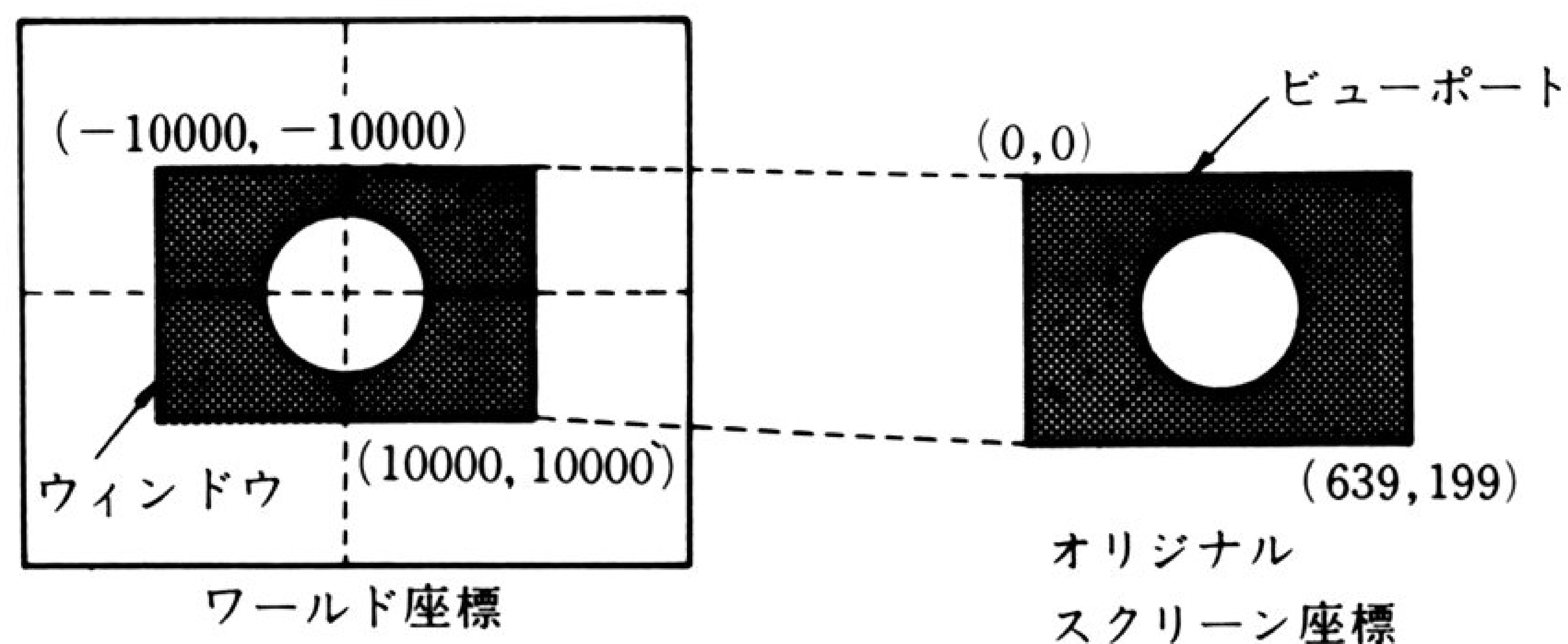
```
window（0，0）－（639,199）
view（0，0）－（639,199）
```

・ウィンドウの内部には円がありません．写しだす対象がウィンドウの外なので，画面に何もみえません．




```
screen　0，0⏎
window（－10000，－10000）－（10000，10000）⏎
circle（0，0），5000，1⏎
```

・ウィンドウが大きくなり，内部に円を含んでいます．ディスプレイ上に円が表示されます．



(8) ワールド座標値の制限

circle文やline文などのグラフィック命令で使用するワールド座標の座標値には次の様な制限があります．

window $(a_1, b_1) - (a_2, b_2)$

view $(x_1, y_1) - (x_2, y_2)$

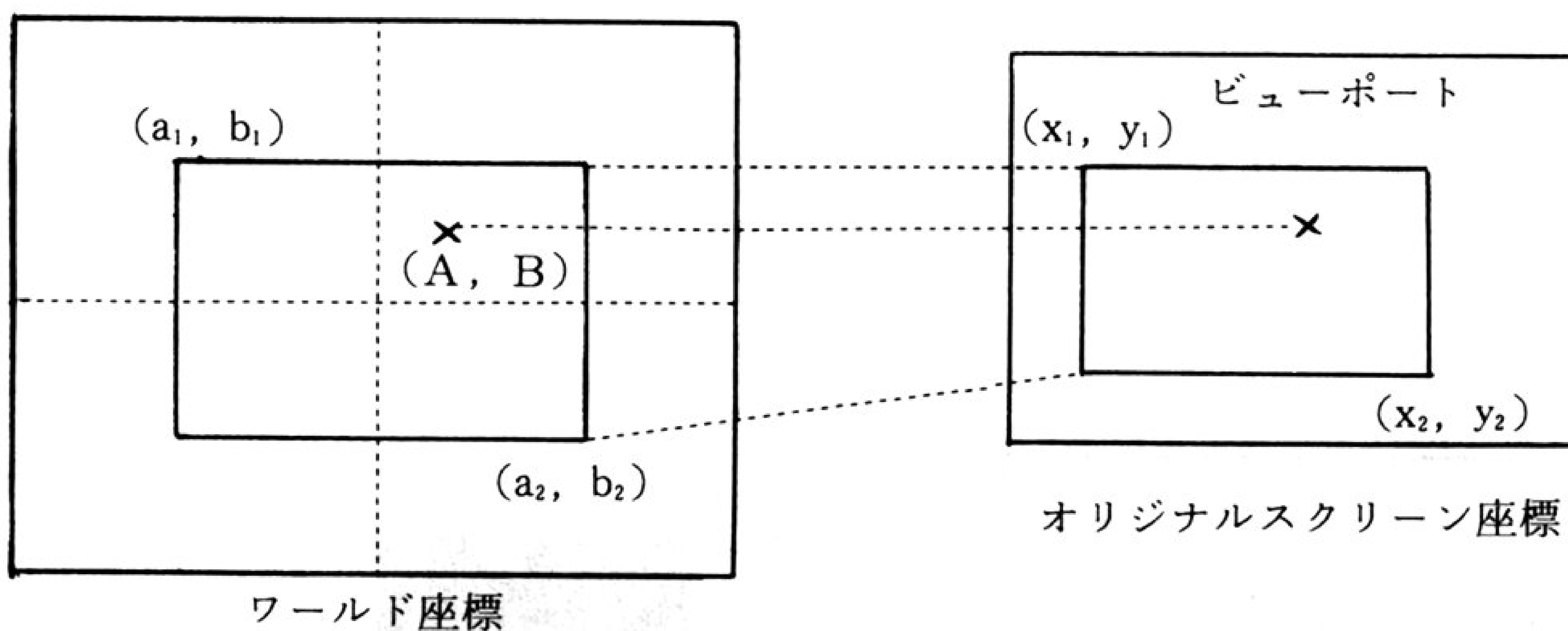
の時，ワールド座標値（A，B）は次の制限があります．制限値を越えた場合はオーバフローエラーになります．

横方向

$$(-32768 - x_1)\frac{a_2 - a_1}{x_2 - x_1} + a_1 \leqq A \leqq (32767 - x_1)\frac{a_2 - a_1}{x_2 - x_1} + a_1$$

縦方向

$$(-32768 - y_1)\frac{b_2 - b_1}{y_2 - y_1} + b_1 \leqq B \leqq (32767 - y_1)\frac{b_2 - b_1}{y_2 - y_1} + b_1$$



例 window（0，0）－（639，399）

view （0，0）－（639，399） の時

$$(-32768 - 0)\times\frac{639-0}{639-0} + 0 \leqq A \leqq (32767 - 0)\times\frac{639-0}{639-0} + 0$$

$$(-32768 - 0)\times\frac{399-0}{399-0} + 0 \leqq B \leqq (32767 - 0)\times\frac{399-0}{399-0} + 0$$

したがって

$$-32768 \leqq A \leqq 32767$$

$$-32768 \leqq B \leqq 32767$$

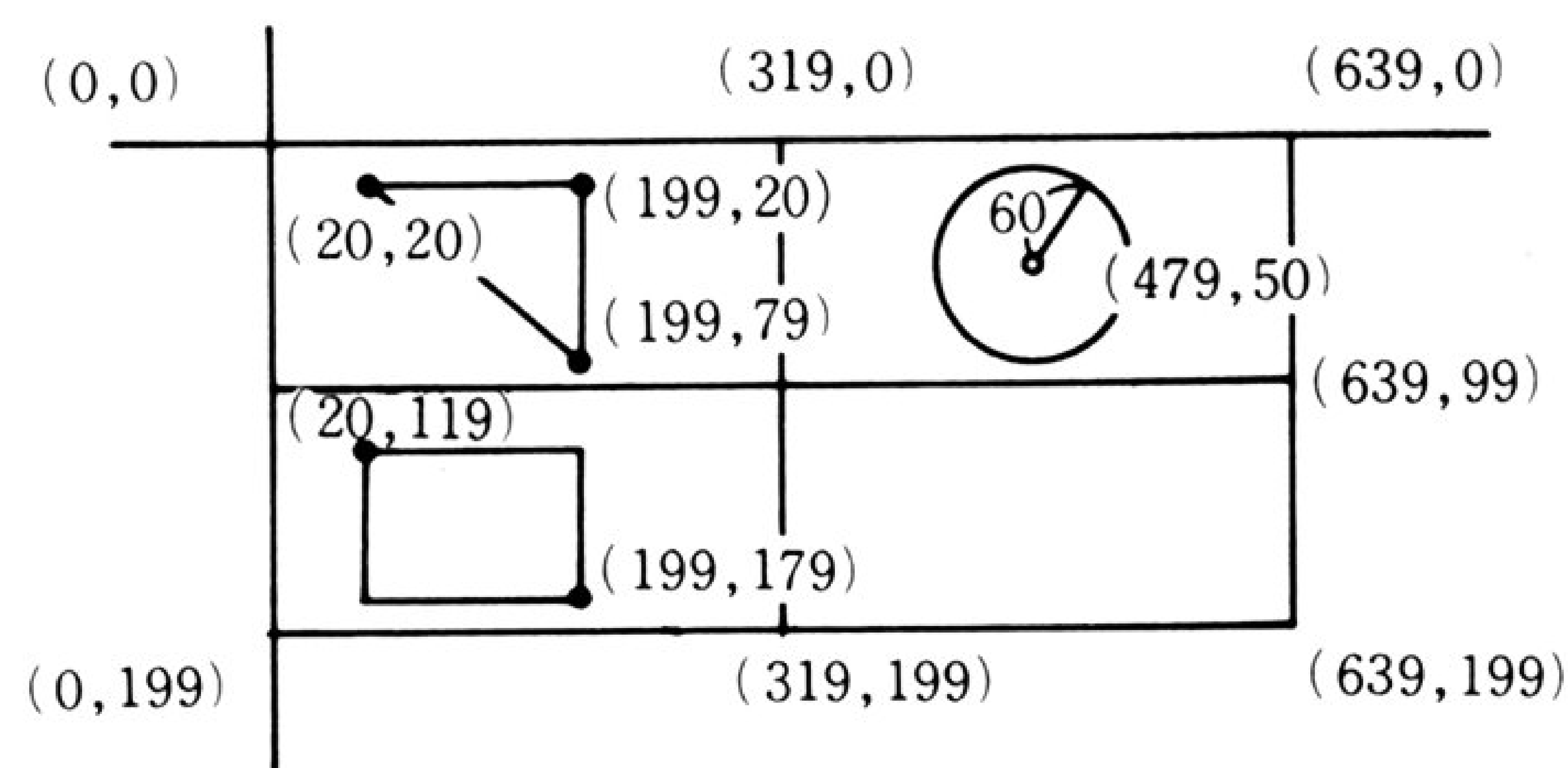
8.1.2 図形の移動，拡大，縮小

ワールド座標系で，プログラムした図形をディスプレイ画面に出力してみます．出力された画面上の図形を，プログラムを変更しないで，移動，拡大，縮小が，ウィンドウ，ビューポートの操作で容易にできます．

簡単な例で説明します．

(1) プログラムした図形

・下記の図は，三角形，円，長方形の3つの図形をワールド座標系で描こうとしたものです．

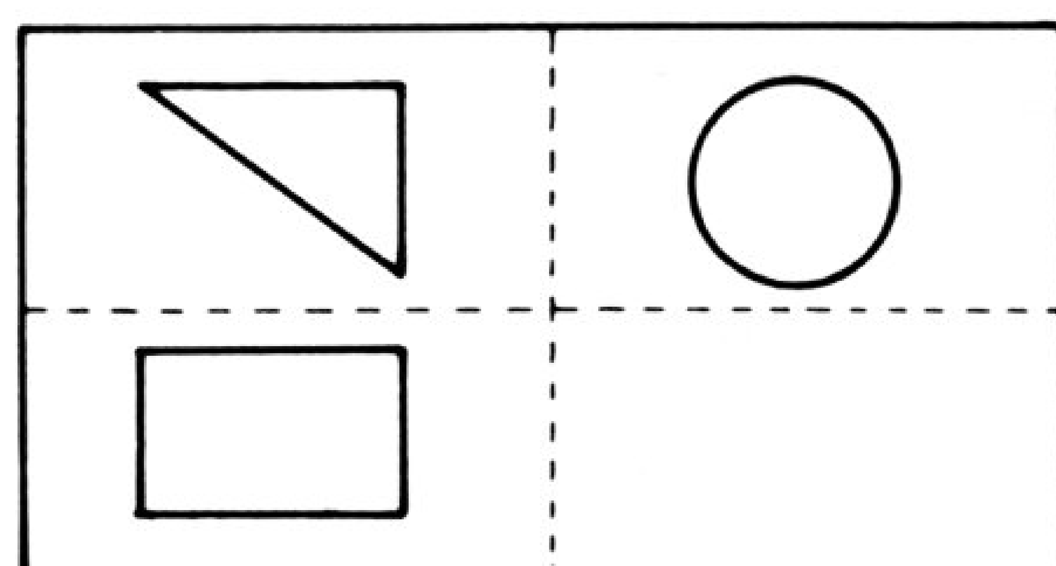


ワールド座標系

・同じような配置でディスプレイするつもりなので，このような座標をとることにします．

```
40    *disp
50    circle (479, 50), 60, 6
60    line (20, 20) - (199, 20), 1 : line (199, 20) - (199, 79), 1
70    line (20, 20) - (199, 79), 1 : line (20, 119) - (199, 179), 2, B
80    return
```

(2) プログラム上の配置と同じ図形を表示します．

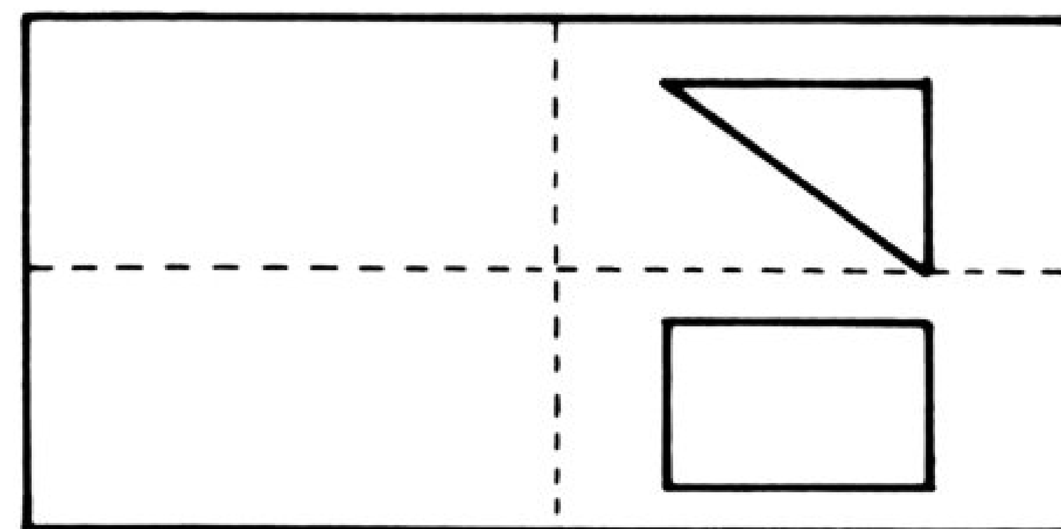


ディスプレイ画面

```
10    screen 0, 0
20    gosub *disp
30    end
```

```
run ⏎
```

(3) 図形を右へ移動してみます.

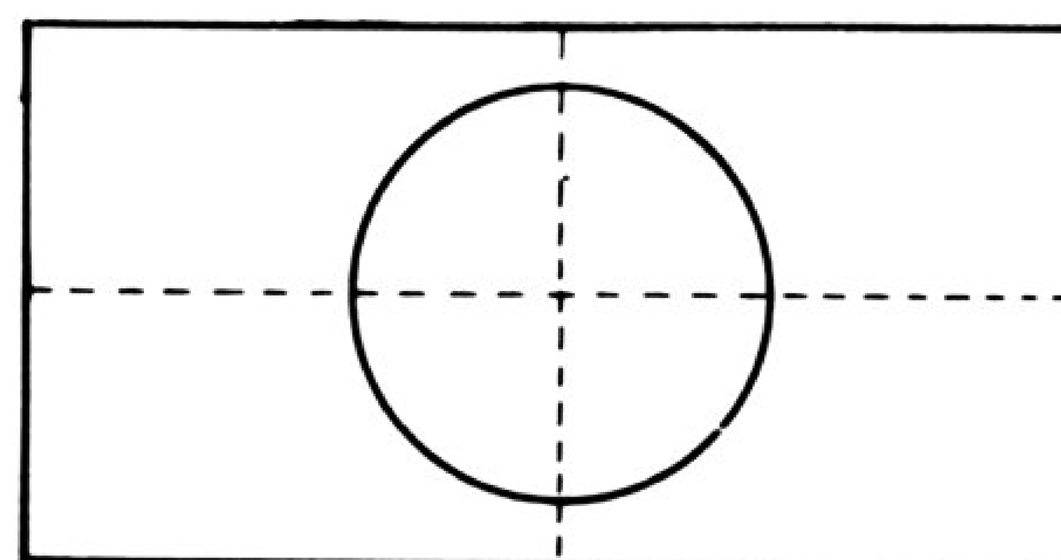


ディスプレイ画面

```
 90    window (−319, 0) − (319, 199)
100    view (0, 0) − (639, 199), 0, 7
110    gosub ＊disp←(1)の図形
120    end
```

run 90 ⏎

(4) 円の部分を拡大してみます.

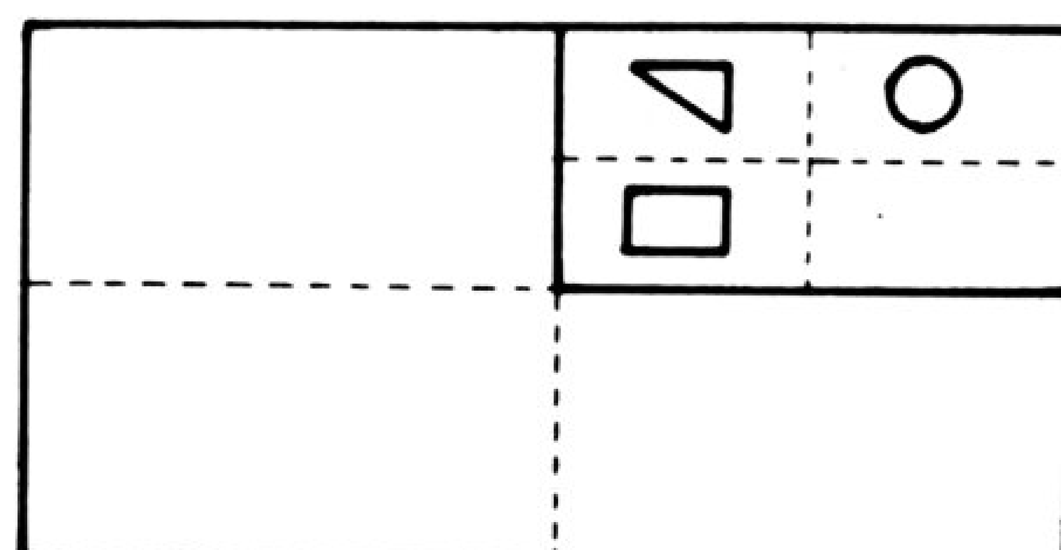


ディスプレイ画面

```
130    window (320, 0) − (639, 99)
140    view (0, 0) − (639, 199), 0, 7
150    gosub ＊disp←(1)の図形
160    end
```

run 130 ⏎

(5) 全体を 1 / 4 に縮小します.

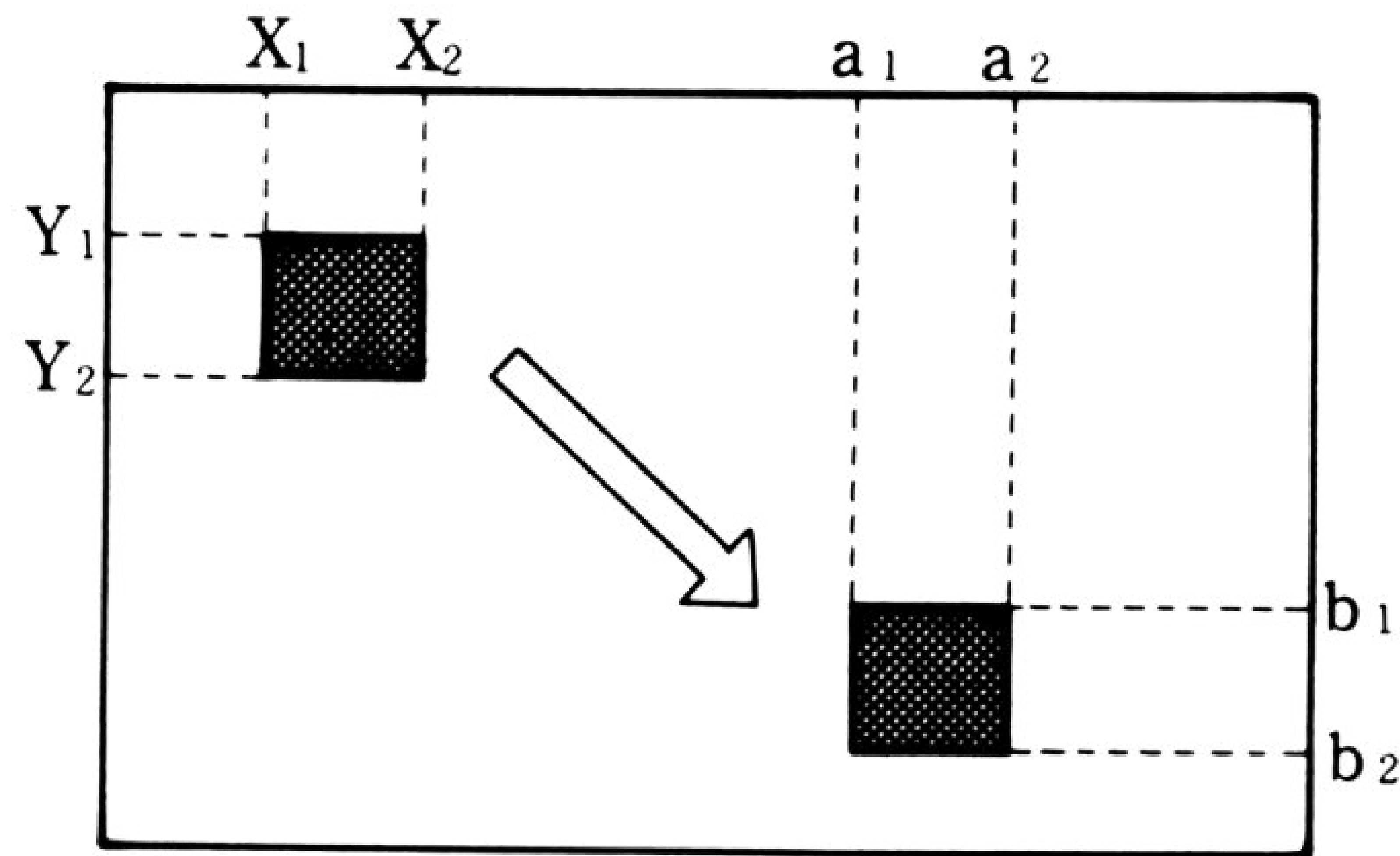


ディスプレイ画面

```
170    cls 2
180    window (0, 0) − (639, 199)
190    view (320, 1) − (638, 99), 0, 7
200    gosub ＊disp←(1)の図形
210    end
```

run 170 ⏎

図形の移動はget@文およびput@文を使用しても行えます.



この例のように（X_1, Y_1）と（X_2, Y_2）で指定される領域に表示されている図形を，（a_1, b_1）と（a_2, b_2）で指定される領域に表示したい場合は，次の命令を実行します．（座標はスクリーン座標です．）

```
get@(X1, Y1)-(X2, Y2), A
put@(a1, b1), A, or
```

get@文は，指定された領域の画面上のデータを配列Aに読み込みます．

put@文は，逆に，配列Aに記憶されている画面のデータを指定された位置に表示します．（上の例ですと，（X_1, Y_1）が（a_1, b_1）に対応します．（a_2, b_2）は，自動的に決まります．）もちろん，配列Aは，get@文を実行する前にdim文で宣言していなければなりません．

注　意　put@文では，配列内のグラフィックパターンと，すでにある画面上のグラフィックパターンを1ビット（ドット）ごとにor（論理和），xor（排他的論理和），and（論理積）し，その結果を画面に表示します．

(2)　get@文，put@文で使用する配列の大きさ

get@文，put@文を実行するのに必要な配列の大きさは，次の式で求めることができます．

？（int（（横のドット数＋7）￥8）＊a＊（縦のドット数）＋4）￥b＋1 [↵]

int　：小数点以下を四捨五入して整数にまとめる関数

a　：4　16色カラーモードの場合
　　3　8色カラーモードの場合
　　1　白黒モードの場合

b　：8　倍精度実数型配列を使用する場合
　　4　単精度実数型配列を使用する場合
　　2　整数型配列を使用する場合

￥　：　整数の除算で商は小数点以下が切り捨てられた整数となります．

例　8色カラーモードでget@(10, 10)-(29, 39), D%を実行する場合
（D%は，整数型配列を表わしています．）

a＝3，横のドット数＝20，縦のドット数＝30，b＝2（Dが整数型配列だから）

？（int（（20＋7）￥8）＊3＊30＋4）￥2＋1 [↵]

〔＝（3＊3＊30＋4）/2＋1＝274/2＋1＝138〕

したがって配列の宣言は，

dim D% (137)

となります．(配列の添字は，通常は0から始まります)

8.2 カラー指定

「7.2 グラフィック画面」でも述べましたが，N88-BASIC (86) のグラフィック機能には2つのモードがあります．

(1) 基本グラフィックモード

(2) 拡張グラフィックモード

この2つのモードはシステムの起動時に一意に決まるものであり，切り換えながら使用していくものではありません．

(1) 基本グラフィックモード

ROMモードBASIC使用時は無条件にこのモードになります．

また，DIP SWの拡張グラフィックスイッチがOFFの場合，N88-日本語BASIC(86) システムディスクからシステムを立上げても基本グラフィックモードになります．

(2) 拡張グラフィックモード

DIP SWの拡張グラフィックスイッチがONの場合，N88-日本語BASIC(86) システムディスクからシステムを立上げると，拡張グラフィックモードになります．

基本グラフィックモードと拡張グラフィックモードの相違点は次のとおりです．

	基本グラフィックモード	拡張グラフィックモード
カラー指定	システムが規定した8色の色の中から8色が選べます．このモードを8色中・8色モードと呼びます．	8色中・8色モード 4096色中・8色モード注(1) 4096色中・16色モード注(1) の3つのモードを選ぶことができます． 4096色中・8色モードおよび4096色中・16色モードは中間色が使用できるモードです．
高速描画機能	高速描画のためのハードウエアは使用しません．	高速描画のためのハードウエアを描画時に適用します．
利用者メモリの圧迫	利用者メモリは特に圧迫しません．	22KBの拡張グラフィック手続きがメモリにロードされます．その分利用者メモリが減ります．

注(1) 4096色中・8色モードおよび4096色中・16色モードは中間色が表示できるモードですが，接続されているディスプレイ装置がアナログRGB対応ディスプレイの場合に限り正しく表示できます．

ここでは，基本グラフィックモードと拡張グラフィックモードで特に機能が異なるカラー指定について説明します．

8.2.1 パレットモード

(1) パレット番号とカラーコード

グラフィック画面に対する色を定めたり，変更する場合に，その都度，色を指定している沢山のステートメントを変更することは大変です．パレットを使うと，パレットと色との対応指定を変更するだけで，いっせいに図形の色が変ります．プログラムを作成する場合に絶対的な色を気にするよりは，図形全体の色のバランス（配色）を相対的に考えてゆく方が，柔軟性があり，工夫しやすいという利点があります．

例えば，次のようなことができます．

```
screen 0, 0 ⏎
circle (100, 100), 50, 1 ⏎
```

これで青い円が描けます．この命令を変えないで，パレットを変更することによって，円を赤くすることもできます．

```
color=（1， 2）⏎
```

もう一度，青い円にもどります．

```
color=（1， 1）⏎
```

このように，現在グラフィック画面に表示されている色およびこれからグラフィック画面に対しておこなう表示要求（たとえば，CIRCLE文による円の描画）における色は，指定したパレットに現在割り付けられている色で決定されます．パレットにはパレット番号が付けられており，どのパレットを選択するかはパレットに付けられた番号,すなわちパレット番号でおこないます．

また，色には色の種類を表わす数値が対応付けられており，これをカラーコードと呼びます．

(2) パレットモード

使用できるパレット番号とカラーコードの種類はパレットモードによって異なります．

パレットモードには次の3種のモードがあり，各モードにおいて，使用できるパレット番号とカラーコードの値が異なります．

・8色中・8色モード

このモードは8個のパレットにシステムが決めた8色を対応付けることができます．

・4096色中・8色モード

このモードでは4096色中の任意の8色を8個のパレットに対応付けることができます．

・4096色中・16色モード

このモードでは4096色中の任意の16色を16個のパレットに対応付けることができます．

基本グラフィックモードでは8色中・8色モードしか使用できません．

拡張グラフィックモードではすべてのモードを使用することができます．

パレットモードは次の命令で指定します．

COLOR〔〈ファンクションコード〉〕〔,〈バックグラウンドカラー〉〕〔,〈ボーダーカラー〉〕〔,〈フォアグラウンドカラー〉〕〔,〈パレットモード指定〉〕

①〈ファンクションコード〉

このパラメータはグラフィック画面に関係ないのでここでは特に説明しません．「BASICリファレンスマニュアル〔1〕COLOR」を参照してください．

②〈フォアグラウンドカラー〉

グラフィック画面に，点や線を表示したりするときに，使われる色のことです．グラフィック命令で特別に色指定をしないと，フォアグラウンドカラーが採用されます．

③〈バックグラウンドカラー〉

グラフィック画面の背景の色のことです．clsで画面をクリアすると，画面の背景がこの色に変ります．また，preset文を，色指定なしで実行すると，この色が採用されます．

④〈ボーダーカラー〉

ディスプレイ画面上で，BASICによって使うことのできる，エリアの外枠の色のことです．ボーダーカラーの指定は，標準カラーディスプレイの場合に使用できます．高解像度ディスプレイを使用する場合，意味を持ちません．

注 意　〈フォアグラウンドカラー〉〈バックグラウンドカラー〉は「パレット番号」で指定します．

〈ボーダーカラー〉に対してはカラーコードを直接指定します．

⑤〈パレットモード指定〉

〈パレットモード指定〉はグラフィック画面に対する色指定のモードを指定します．

〈パレットモード指定〉に指定できる値とその意味は次の通りです．

0：8個のパレットにシステムが決めた8色を対応づけるモードです（このモードを8色中・8色モードと呼びます）

1：8個のパレットに4096色中の任意の8色を対応づけるモードです（このモードを4096色中・8色モードと呼びます）

2：16個のパレットに4096色中の任意の16色を対応づけるモードです（このモードを4096色中・16色モードと呼びます）

システムが立上がった直後のパレットモードは8色中・8色モードです．以降，〈パレットモード指定〉が指定された場合に限りパレットモードが切り換わります．

各モードにおけるパレットとカラーコードの対応づけの方法に関しては8.2.2および8.2.3を参照してください．

〈パレットモード指定〉がある場合，パレットモードが切り換わりパレットとカラーコードの関係は次のように初期化されます．

8色中・8色モード			4096色中・8色モード			4096色中・16モード		
〈パレット番号〉		〈カラーコード〉	〈パレット番号〉		〈カラーコード〉	〈パレット番号〉		〈カラーコード〉
0	←	0（黒）	0	←	&H000（黒）	0	←	&H000（黒）
1	←	1（明るい青）	1	←	&H00F（明るい青）	1	←	&H00F（明るい青）
2	←	2（〃　赤）	2	←	&H0F0（〃　赤）	2	←	&H0F0（〃　赤）

3 ← 3 （ 〃 紫）	3 ← &H0FF （ 〃 紫）	3 ← &H0FF （ 〃 紫）
4 ← 4 （ 〃 緑）	4 ← &HF00 （ 〃 緑）	4 ← &HF00 （ 〃 緑）
5 ← 5 （ 〃 水色）	5 ← &HF0F （ 〃 水色）	5 ← &HF0F （ 〃 水色）
6 ← 6 （ 〃 黄色）	6 ← &HFF0 （ 〃 黄色）	6 ← &HFF0 （ 〃 黄色）
7 ← 7 （白）	7 ← &HFFF （白）	7 ← &HFFF （白）
		8 ← &H777 （灰色）
		9 ← &H00A （少し暗い青）
		10 ← &H0A0 （ 〃 赤）
		11 ← &H0AA （ 〃 紫）
		12 ← &HA00 （少し暗い緑）
		13 ← &HA0A （ 〃 水色）
		14 ← &HAA0 （ 〃 黄色）
		15 ← &HAAA （ 〃 白）

注　意

基本グラフィックモードにおいては，８色中・８色モードしか使用できません．
〈パレットモード指定〉は必ず省略してください．
拡張グラフィックモードにおいては，すべてのモードを選択することができますが，4096色中・８色モード及び4096色中・16色モードはアナログRGB対応ディスプレイが接続されている場合のみ有効です（エラーにはなりませんがアナログRGB対応ディスプレイが接続されていない場合，指定通りの色は出ません）．

8.2.2　基本グラフィックモードにおけるカラー指定

前節で説明した通り，基本グラフィックモードにおいては８色中・８色モードしか使用できません．

グラフィック画面への色の指定はすべてカラーパレットによっておこないます．

カラーパレットに対する色の設定はCOLOR文で次のようにおこないます．

COLOR＝（〈パレット番号〉，〈カラーコード〉）

この命令は，どのパレットにどの色を対応させるか決めるものです．

〈パレット番号〉とは，ユーザーのために用意された０から７までの８つのカラーパレットにつけられた固有の番号でユーザーはそれぞれのパレットにどの色を持ってくるかを〈カラーコード〉によって指定します．基本グラフィックモードにおけるパレット番号とカラーコードの関係は次の通りです．中間色の指定はおこなえません．

〈パレット番号〉		〈カラーコード〉
0		0 （黒）
1	どのパレットに	1 （青）
2	どのカラーコード	2 （赤）
3	を対応させるか任	3 （紫）
4	意に決めることが	4 （緑）
5	できます．	5 （水色）
6		6 （黄色）
7		7 （白）

たとえば，COLOR＝（２，４）を実行した場合，パレット２にはカラーコード４（緑）が割り付けられたことになります．

システムが起動された初期状態では，カラーパレットにはそのパレット番号と同じカラーコードが割り付けられています．したがって，もしカラーパレットを全く変更しなければ，カラーパレットとカラーコードは同じものとして使ってさしつかえありません．

カラーパレット変更後は，グラフィック画面に表示されている色及びこれから画面に対して行うすべてのグラフィック命令（PSET, LINEなど）におけるカラー表示は，そのパレット番号に割り付けられているカラーコードでおこなわれます．

したがってプログラム中でカラーパレットを変更した場合は，困惑しないようにプログラム終了時にパレットを初期状態に戻しておくほうが良いでしょう（基本グラフィックモードにおいては，パレットを初期状態に戻すための専用命令は特に用意されていません．したがって，COLOR文を使用し，各パレットに初期カラーコードを１つずつ割り当てなおさなければなりません．なお，拡張グラフィックモードにおいてはパレット初期化用に専用命令が用意されています)．
パレットの使用が許されているのは，グラフィック画面に対してだけです．

8.2.3 拡張グラフィックモードにおけるカラー指定

基本グラフィックモードにおけるカラー指定と同様にグラフィック画面に対する色の指定はすべてカラーパレットによっておこないます．

カラーパレットに対する色の設定はCOLOR文でおこないます．

COLOR＝〔(〈パレット番号〉，〈カラーコード〉)〕

〈パレット番号〉とは０から７（あるいは０～15）の８個（あるいは16個）のカラーパレットにつけられた固有の番号で，それぞれのパレットにどの色を対応づけるかを〈カラーコード〉によって指定します．

パレットの個数及びカラーコードの値はパレットモードによって異なります（パレットモードに関しては前節8.2.1で説明しています)．

各パレットモードにおける〈パレット番号〉と〈カラーコード〉の関係は次の通りです．

８色中・８色モード

〈パレット番号〉		〈カラーコード〉	
0		0 （黒）	
1	どのパレットに	1 （青）	
2	どのカラーコード	2 （赤）	の８種類の任意の８個

3	を対応させるか任	3 （紫）	（同じカラーコードを複数のパレット
4	意に決めることが	4 （緑）	に対応づけてもかまいません．）
5	できます．	5（水色）	
6		6（黄色）	
7		7 （白）	

4096色中・8色モード

〈パレット番号〉		〈カラーコード〉	
0		&H000	
1	どのパレットに		の4096色中の任意の8個
2	どのカラーコード		（同じカラーコードを複数のパレット
3	を対応させるか任		に対応づけてもかまいません．）
4	意に決めることが		
5	できます．		
6			
7		&HFFF	

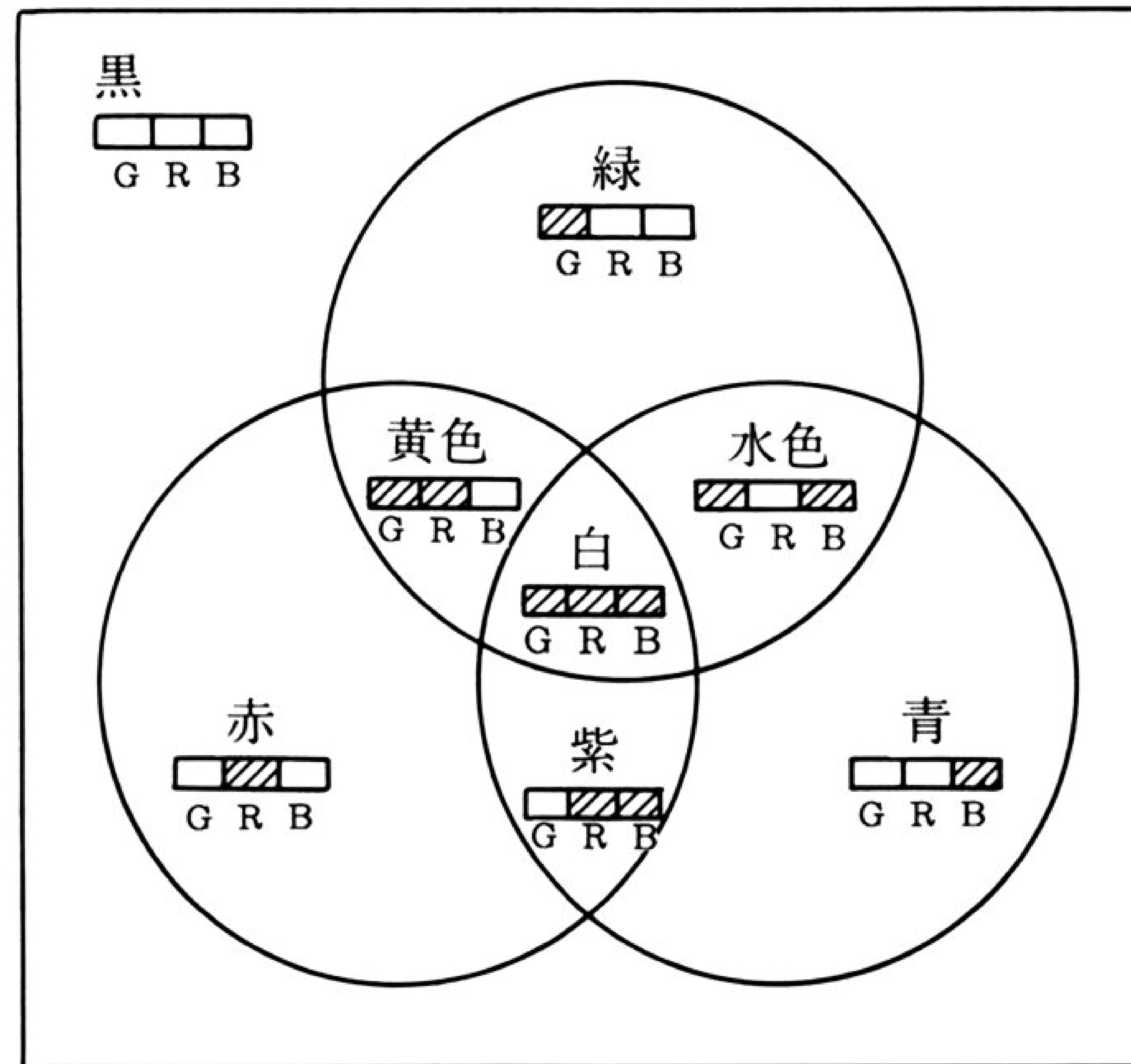
4096色中・16色モード

〈パレット番号〉		〈カラーコード〉	
0		&H000	
1			
2			
3			
4			
5	どのパレットに		
6	どのカラーコード		
7	を対応させるか任		の4096色中の任意の16個
8	意に決めることが		（同じカラーコードを複数のパレット
9	できます．		に対応づけてもかまいません．）
10			
11			
12			
13			
14			
15		&HFFF	

ここで4096色中・8色モード及び4096色中・16色モードにおけるカラーコードの値と色の関係について説明します．

(1) 色のしくみ

グラフィック画面に表示できる色はすべて基本色である緑（G），赤（R），青（B）の組み合わせで表現されます．



▨ はONの状態を，□ はOFFの状態を示しています．

たとえば，水色の場合 ▨□▨ (G R B) は，BとGがONの状態です．すなわち水色は青と緑を混ぜて表現される色であることを示しています．

(2) 4096種類の色

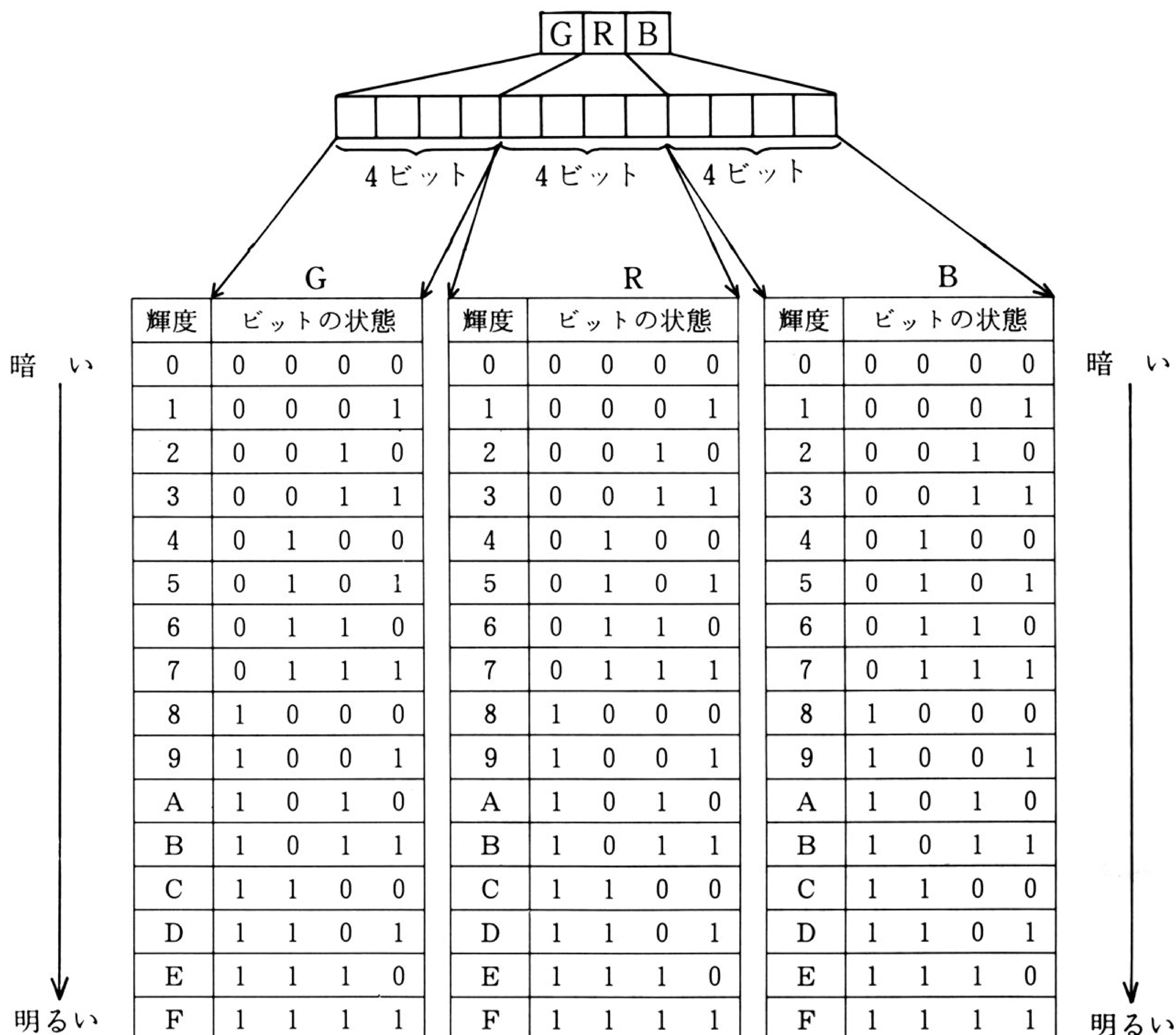
黒，青，赤，紫，緑，黄色，水色，白の色のしくみは(1)で示した通りですが，オレンジ色や黄緑などの中間色はどのように表現するのかを説明します．

たとえば，オレンジ色というのは，黄色に赤を混ぜた色です．黄色は赤と緑を混ぜた色ですので，それにさらに赤色を混ぜるとオレンジ色になります．つまり，赤と緑を混ぜ合わせる際に，緑よりも赤を多く混ぜるとオレンジ色が表現できます．赤と緑を混ぜ合わせる割り合いは，「輝度」によって調整することができます．

「輝度」はその名の通り明るさの度合いを示したものです．

G,R,Bはそれぞれ16段階の輝度を表現することができます．

G,R,Bはそれぞれ4ビットの情報を持ち，このビットの状態で輝度を表わしています（16段階の輝度を表わすのには16進表現が便利です）．



G

輝度	ビットの状態			
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
A	1	0	1	0
B	1	0	1	1
C	1	1	0	0
D	1	1	0	1
E	1	1	1	0
F	1	1	1	1

R

輝度	ビットの状態			
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
A	1	0	1	0
B	1	0	1	1
C	1	1	0	0
D	1	1	0	1
E	1	1	1	0
F	1	1	1	1

B

輝度	ビットの状態			
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
A	1	0	1	0
B	1	0	1	1
C	1	1	0	0
D	1	1	0	1
E	1	1	1	0
F	1	1	1	1

輝度が0というのは，その色が表示されないことを意味します．

オレンジ色を表現する場合は，輝度が&H9の緑（暗い緑）と輝度が&HFの赤（最も明るい赤）を混ぜ合わせることによって表現します（カラーコードに&H9F0と指定します）．

このように輝度を変えることで，様々な色合いが表現できます．

また，基本の7色についてもその輝度を変えれば，暗い赤や暗い黄色などの表現ができます．

　例）明るい黄色→輝度が&HFの緑と&HFの赤の混ぜ合わせ

　　　暗い黄色→輝度が&H7の緑と&H7の赤の混ぜ合わせ

緑（G）と赤（R）と青（B）はそれぞれ16段階の輝度が表現できます．そして，これらの色の混ぜ合わせを自由に行うと16×16×16＝4096種類の色が表現できます．

拡張グラフィックモードにおいてはカラーパレットとカラーコードの対応を各パレットモードにおける初期状態にリセットすることができます．

次に示すように“＝”（等号）も含めて〈パレット番号〉および〈カラーコード〉を省略しま

す.

COLOR ⏎

各パレットモードにおけるパレットの初期状態に関しては,「8.2.1　パレットモード」を参照してください.

8.3　タイリング

タイリングは主に8色中・8色モードで中間色を表現するために使用されるテクニックです.

もちろん,4096色中・8色あるいは16色モードにおいて使用することもできます.この場合,より複雑な模様を生成するのに役立ちます.

タイリングはPAINT文およびCIRCLE文,LINE文等のぬりつぶし指定の時に使用できます.

この章ではPAINT文を例に8色モードにおけるタイリングの仕組みおよび方法を説明します.

16色モードにおけるタイリングの考え方については「BASICリファレンスマニュアルPAINT文」を参照してください.

paint文は,指定した色の枠内をある色で塗ることができる命令です.

(1)　パレット番号に従ってペイントする場合

パレット番号で指定できる色でペイントする場合は,次のpaint文を使用します.

paint (x,y),〈領域色のパレット番号〉,〈境界色のパレット番号〉

ここでは,(x,y)は塗り始めるドットのワールド座標です.

例　赤い枠の中を水色でペイントする場合.

```
screen    0, 0 ⏎
cls    2 ⏎
line (50, 50) - (450, 150), 2, b ⏎
line (100, 75) - (200, 100), 2, b ⏎
paint (300, 100), 5, 2 ⏎
```

(2)　タイルストリングでペイントする場合(これをタイリングと呼びます)

タイリングの場合は,ドットごとに色の指定をおこないます.タイリングの場合のpaint文は次の形になります.

paint (x,y),〈タイルストリング〉,〈境界色のパレット番号〉

〈タイルストリング〉は,文字列で与えます.

〈タイルストリング〉は,通常,次のように指定します.

```
chr$(&hXX)+chr$(&hXX)+chr$(&hXX)
 (青の要素)    (赤の要素)    (緑の要素)
```

chr$は( )のキャラクタコードをもつ文字を与える関数です.&hは以下の数値が16進数であることを示します.まず,実際に中間色を使ってみましょう.

例　白で描かれた円の中をオレンジ色で塗りつぶす場合.

```
screen    0,0 ⏎
cls    2 ⏎
tile$=chr$(&h00)+chr$(&hff)+chr$(&h55) ⏎
```

```
circle(320,100),100,7 ⏎
paint(320,100),tile$,7 ⏎
```

この例では，〈タイルストリング〉をtile$という文字変数に代入してから，paint文を実行しています．実際に，ディスプレイの画面を各ドットで考えてみましょう．

	2進数	16進数
青の要素	0 0 0 0 0 0 0 0	0 0
赤の要素	1 1 1 1 1 1 1 1	F F
緑の要素	0 1 0 1 0 1 0 1	5 5

0はその要素がないことを表わし，1はその要素があることを表わしています．ですから，ディスプレイ上の8ドットを取りあげると，

青の要素	0	0	0	0	0	0	0	0	(00)
	+	+	+	+	+	+	+	+	
赤の要素	1	1	1	1	1	1	1	1	(FF)
	+	+	+	+	+	+	+	+	
緑の要素	0	1	0	1	0	1	0	1	(55)
	‖	‖	‖	‖	‖	‖	‖	‖	
	赤	黄	赤	黄	赤	黄	赤	黄	

となります．この8ドットをずっと並べて表示しますと，赤と黄が混ざるので，あたかもオレンジ色のように見えます．先程の例の“00”，“ff”，“55”を適当な16進数に変えて入力してみると，いろいろな色や模様が表示できます．

ここまでは，8ドットパターンを1行だけ指定して，それを繰り返して色を塗っていたわけですが，複数の行のドットパターンを指定すると，もっと複雑な模様も表示できます．先程の例を使いますと，

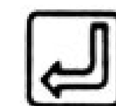
```
tile$=chr$(&hXX)+chr$(&hXX)+chr$(&hXX)
tile$=tile$+chr$(&hXX)+chr$(&hXX)+chr$(&hXX)
tile$=tile$+chr$(&hXX)+chr$(&hXX)+chr$(&hXX)
        ⋮
tile$=tile$+chr$(&hXX)+chr$(&hXX)+chr$(&hXX)
```

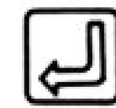
というように，ドットパターンを指定します．

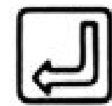
例　白い円の中を模様で塗りつぶす場合

```
screen    0,0 ⏎
cls       2 ⏎
```

```
tile$=chr$(&h12)+chr$(&h34)+chr$(&h56)
```


```
tile$=tile$+chr$(&h78)+chr$(&h9a)+chr$(&hbc)
```


```
tile$=tile$+chr$(&hde)+chr$(&hf0)+chr$(&h12)
```


```
circle(320, 100), 100, 7
```


```
paint(320, 100),tile$, 7
```


(3) 白黒モードあるいは高分解能白黒モードにおけるタイリング

この場合は，青，赤，緑の各要素の指定が1つになり，

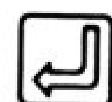
chr$(&hXX)

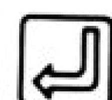
で1行分のドットパターンを指定します．このとき，1ならば白，0ならば黒が表示されます．

例

&haa	1	0	1	0	1	0	1	0
	‖	‖	‖	‖	‖	‖	‖	‖
	白	黒	白	黒	白	黒	白	黒

```
screen    1, 0, 0, 1
```


```
cls    2
```


```
tile$=chr$(&haa)
```


```
circle(200, 100), 80, 1
```


```
paint(200, 100),tile$, 1
```


第9章

画面ハードコピー

ディスプレイ画面に表示された画面のイメージをそのままプリンタに出力することができます．その特徴は次の通りです．

・テキスト画面の情報だけを選んで出力することができます．
・グラフィック画面の情報だけを選んで出力することができます．
・テキスト画面の情報とグラフィック画面の情報を合成して出力することができます．
・PC-PR201V系カラープリンタが接続されている場合，画面イメージ通りのカラー出力も行えます．

9.1 N88-BASIC (86)･･･ROM BASICモード使用時の画面ハードコピー

ROM BASICモード時の画面ハードコピー機能は次の通りです．

項番	キー操作	COPY文	機　　能
1	CTRL + COPY	COPY 1	テキスト画面のみの画面ハードコピーを行います．出力される文字は接続されているプリンタの印字体となります．たとえば，明朝体印字が可能なPC-PR201系プリンタを使用しますと明朝体による印字となります．
2	GRPH + COPY	COPY 2	グラフィック画面のみの画面ハードコピーを行います．
3	COPY	COPY 3	テキスト画面とグラフィック画面を重ねてプリンタに出力します．(項番1と項番2の出力情報をそのまま重ねたようなイメージです)
4	――	COPY 4	グラフィック画面の縮小コピー（縦方向に縮小します．グラフィック画面が640×200ドットモードの場合，縦方向に縮小された形で出力されますので，グラフィック画面に表示した漢字等が見易くなります．
5	――	COPY 5	項番3の両画面コピーをそのまま縦方向に縮小したコピーです．項番4と同様，640×200ドットモード時に有効です．

注 意 (1) 項番4，5の機能はキーによる指定は行えません．

(2) PC-PR201系プリンタを接続している場合，メモリスイッチSW5・2^0ビットをONにして下さい（このスイッチは標準状態でONになっていますが，何らかの理由でOFFにした場合，システムディスクに格納されているユーティリティ「switch.n88」を使用して設定し直して下さい）．

9.2 N88-日本語 BASIC (86)・・・DISK BASICモード使用時の画面ハードコピー

(1) 通常の画面ハードコピー

特にメモリスイッチの設定をしない限り，ROM BASICモードと同一の画面ハードコピー機能となります．

(2) 拡張画面ハードコピー

メモリスイッチSW6・2^4がONの状態でN88-日本語BASIC (86)を起動した場合，拡張画面ハードコピー機能が使用できます（メモリスイッチの設定には，システムディスクに格納されているユーティリティ「switch. n88」を御使用ください）．メモリスイッチにより拡張画面ハードコピー機能の使用を宣言しますと，拡張画面ハードコピー手続きが利用者メモリ領域にロードされます（約6 KBを占めます）．その分，変数エリア，配列データエリア等の領域が小さくなりますので御注意下さい．

(a) PC-PR201V系カラープリンタを使用する場合

・メモリスイッチSW5・2^3ビットがONの場合（メモリスイッチの設定には，システムディスクに格納されているユーティリティ「switch. n88」を御使用ください）

カラー画面ハードコピーを行います．

項番	キー操作	COPY文	機　　能
1	CTRL ＋ COPY	COPY 1	テキスト画面のみのカラー画面ハードコピーを行います．
2	GRPH ＋ COPY	COPY 2	グラフィック画面のみのカラー画面ハードコピー（白黒反転）を行います
3	COPY	COPY 3	両画面を合成したイメージのカラー画面ハードコピー（白黒反転）を行います
4	——	COPY 4	グラフィック画面のみのカラー画面ハードコピー（白黒反転せず）を行います
5	——	COPY 5	両画面を合成したイメージのカラー画面ハードコピー（白黒反転せず）を行います

注 意 4096色中・8色，4096色中・16色モードでのハードコピーはモノクロのハードコピーを行います．

また，通常の画面ハードコピーと同様メモリスイッチSW5・2^0ビットをONにして下さい．

・メモリスイッチSW5・2^3ビットがOFFの場合（メモリスイッチの設定には，システムディスクに格納されているユーティリティ「switch. n88」を御使用ください）

モノクロ画面ハードコピーを行います．

項番	キー操作	COPY文	機　　能
1	CTRL ＋ COPY	COPY 1	テキスト画面のみの画面ハードコピーを行います
2	GRPH ＋ COPY	COPY 2	グラフィック画面のみの画面ハードコピーを行います
3	COPY	COPY 3	両画面を合成したイメージの画面ハードコピーを行います
4	——	COPY 4	———
5	——	COPY 5	———

注　意　拡張画面ハードコピーにおける両画面を合成したイメージの画面ハードコピー（COPY 3）は通常の画面ハードコピーにおける両画面COPYと異なり，テキスト画面の情報がグラフィック画面に文字通り合成され，ドット出力されます．

(b)　カラープリンタ以外のプリンタ（たとえばPC-PR201系プリンタ）を使用する場合

メモリスイッチSW6・2^4ビットをONにする以外特に準備は必要ありません．なお，PC-PR201系プリンタ使用の場合，通常の画面ハードコピーと同様メモリスイッチSW5・2^0ビットをONにする必要があります．

項番	キー操作	COPY文	機　　能
1	CTRL ＋ COPY	COPY 1	テキスト画面のみの画面ハードコピーを行います
2	GRPH ＋ COPY	COPY 2	グラフィック画面のみの画面ハードコピーを行います
3	COPY	COPY 3	両画面を合成したイメージの画面ハードコピーを行います
4	——	COPY 4	———
5	——	COPY 5	———

注　意　拡張画面ハードコピーにおける両画面を合成したイメージの画面ハードコピー（COPY 3）は通常の画面ハードコピーにおける両画面COPYと異なり，テキスト画面の情報がグラフィック画面に文字通り合成され，ビットで出力されます．

備　考　以上説明した拡張画面ハードコピーの選択条件をまとめると次のようになります．

(1)　まず，拡張画面ハードコピーを選択するために，メモリスイッチSW6・2^4ビットをONにしておく必要があります．

(2)　PC-PR201系プリンタ使用の場合，(1)に加え，メモリスイッチSW5・2^0ビットを

ONにしておく必要があります.

(3) 更に，PC-PR201V系カラープリンタを使用し，カラー画面ハードコピーを出力するメモリスイッチSW5・2^3ビットをONにしておく必要があります.

第10章
入出力装置とファイル

10.1 ファイル

BASICでは，入出力装置との情報のやりとりを「ファイル（書類）」という概念で行います．目的の「ファイル」は，どこの棚にある何という名前のものかを指定しなければなりません．この指定を行うのが，“ファイルディスクリプタ”と呼ばれるものです．ファイルディスクリプタは，棚にあたる〈入出力装置〉を指定する“デバイス名”と〈書類の名前〉にあたる“ファイル名”で構成されています．

(NOTE) 〔ファイルディスクリプタ〕＝〔デバイス名〕＋〔ファイル名〕

10.1.1 デバイス名

デバイス名は，次の表のように，各入出力装置につけられています．

入出力装置名	デバイス名	入力	出力
キーボード	KYBD：	○	×
スクリーン	SCRN：	×	○
プリンタ	LPT1：	×	○
カセットテープ（1200ボー）	CAS1：	○	○
（ 600ボー）	CAS2：	○	○
ディスクドライブ 1	1：	○	○
2	2：	○	○
3	3：	○	○
4	4：	○	○
5	5：	○	○
6	6：	○	○
7	7：	○	○
8	8：	○	○
9	9：	○	○
10	10：	○	○
11	11：	○	○
12	12：	○	○
13	13：	○	○
14	14：	○	○

RS-232Cインタフェース第1回線	COM 1:	○	○
RS-232C 〃 第2回線	COM 2:	○	○
RS-232C 〃 第3回線	COM 3:	○	○

注　意　"KYBD:"と"SCRN:"はDISK BASICモードでだけ指定できます．
また，COM 2，COM 3（RS－232Cインターフェース第2回線および第3回線）を使用する場合，専用のインターフェースボードが必要となります．

デバイス名に含まれる"1"および，ディスクの"1:"は省略できます．ディスクのデバイス名については，「11.2　ディスクにおけるファイルディスクリプタ」を参照してください．

また，ROM BASICモードにおいては，デバイス名を省略すると「cas:（又は，cas 1:）」が指定されたと見なされます．この時，「cas 2:」は省略できません（「2:」と指定してもエラーとなります）．

10.1.2　ファイル名

ファイル名は，最大で「6文字＋拡張子3文字」で構成されています．拡張子は，ファイルの内容を識別するために用います．例えば，プログラムには，6文字以下のプログラム名の後に，拡張子として，"n88"を続けると決めておきます．"test1"というプログラムでしたら，

test1. n88

となります．データファイルでしたら，"dat"を続けると決めておきます．

test. dat

となります．こうしておくと，フロッピィディスクに記録されているファイル名を表示させた場合など，どのファイルが，どんな形式で記録されているのか一目でわかります．しかし，同じ形式のものしか記録しない場合は，9文字まで，プログラム名として使用してもかまいません．

注　意　カセットテープを使用する場合は，「拡張子3文字」は使用できません．すなわち，ファイル名は最大6文字になります．また，RS-232Cポートを入出力装置として選んだ場合は，ファイル名は，RS-232Cポートの仕様を定義するために使用します．詳しくは，「10.4.2　RS-232Cインタフェース」を参照してください．

10.2　ファイルのOPEN, CLOSE

新しいファイルを作成したり，追加・削除するためには，ファイルを開いたり，使い終ったファイルを閉じたりする動作が必要になってきます．この動作を行う命令が，open文とclose文です．open文では，ファイルディスクリプタを指定するだけでなく，そのファイルに対して，どのような処理を行うかを宣言します．この宣言を〈モード〉といいます．

モード	処理
INPUT	既につくられているファイルから，データを読み込みます．
OUTPUT	新しいファイルを作成して書き込みます．
APPEND	既にあるファイルに，続けてデータを書き込みます．

〈モード〉はファイルのポインタの最初の位置を決定します．

INPUT　ポインタの最初の位置はファイルの始めで，ファイルが見つからない場合は，エラーになります．

OUTPUT　最初の位置はファイルの始めで，常に新しいファイルを作ります．

APPEND　最初の位置は，ファイルの終りで，ファイルが見つからない場合は，エラーになります．

for〈モード〉が省略された場合には，ポインタの最初の位置はファイルの先頭になります．ファイルが見つからない場合は，その名前のファイルができます．

ファイル番号について少し説明しておきます．BASICでは，ファイルとの入出力を行うとき，バッファという窓口を設けます．つまり，ファイルにデータを書き込むとき，あるいは逆に読み出すときは，この窓口を通じて行います．この窓口は，同時に一つのファイルしか使用できません．ですから，一度オープンさせた窓口は，一度閉じなければ，他のファイル用にオープンすることはできません．この窓口の番号がファイル番号です．open文でファイルディスクリプタにファイル番号を指定すると，以後は，ファイル番号でファイルを取り扱います．ファイル番号は，♯1から，「How many files?」で指定した値まで指定できます．

（例）　ディスク上に“data”というファイルを作成する場合

```
10  open "1:data" for output as#1
                 :
100  print  #1, "1234"
```

“♯1”がファイル番号で，print♯1では，♯1の窓口，すなわち“data”というディスクファイルに“1234”というデータを出力します．

注　意　フロッピィディスクに対してファイルをOPENしている状態で，そのフロッピィディスクを取り出すと，そのフロッピィディスクの内容を壊してしまう場合があります．フロッピィディスクを取り出す場合は，OPENされたままのファイルがない事を確認してから行ってください．

10.3 ファイルの同時オープン数

システムディスクのIDセクタにオートスタート情報を設定している場合を除いてBASICを立上げると，

How many files (0-15) ?

と尋ねてきます．これは，「同時にいくつのファイルを使用しますか」，または別の表現をすると，

「ファイルとのデータの入出力用に，いくつの窓口（バッファ）を準備すれば良いのですか」と問い合わせてきているのです.同時オープン数は最大15まで指定できます.しかし,ここで「同時に」という点に注意してください．例えば，file 1, file 2, file 3, file 4という4つのファイルを扱うプログラムを使うとします．このとき，file 1とfile 3は，常に使用し，file 2とfile 4は，交互に使用するとします．

この場合はfile 2とfile 4とで1つの窓口を使います．つまりfile 2を使用する場合は，file 2として窓口をオープンし，使い終ったら，すぐにクローズして，同じ窓口をfile 4としてオープンします．そうすれば，4つのファイルを扱うのに3つの窓口で済むわけです．

（例）　How many file(0-15)？3

```
        :
100  open "1 : file 1" for input as #1
200  open "2 : file 2" as #2
300  open "2 : file 3" for input as #3
        :
1000  close #2
1100  open "2 : file 4" for input as #2
```

「ファイルを4つ使うならば，窓口も4つ準備すればよい」と思われる方があるかもしれません．しかし，窓口（入出力用のバッファ）は利用者メモリの中に作られますから，不必要に多く指定すると，それだけ，使えるメモリ領域が少なくなってしまいます．

注　意　How many files?と尋ねてきたとき，0～15の数値以外を指定するか，または，リターンキー（⏎）を入力すると，ディスクユニットが接続されている場合はそのドライブ数が設定されます．ディスクユニットが接続されていない場合は2が設定されます．

10.4　各種入出力装置をファイルとして扱う

BASICにおいては各種の入出力装置をファイルという論理的な装置として取り扱うことができます．

したがって，入出力装置の各種諸元の違いを意識することなく，種々の入出力命令を同じように適用していくことができます．

各種入出力装置とBASICの入出力文の関係は次の通りです．

入出力命令＼デバイス	ディスク n：	CASn：	COM：	KYBD：	SCRN：	LPT：	備　考
BLOAD	○	×	○	×	×	×	○：使用できる ×：使用できない ＊：実際には何も行われない SCRN：とKYBD：とディスクn：は，DISK B-ASICモードでのみ使用可能です． デバイス名と装置の対応は「10.1.1 デバイス名」を参照してください． 各入出力命令の詳細については「B-ASICリファレンスマニュアル」を参照してください．
BSAVE	○	×	○	×	○	○	
CLOSE	○	○*	○	○*	○*	○	
EOF	○	×	○	×	×	×	
FPOS	○	×	×	×	×	○	
GET	○	○	×	○	×	×	
INPUT#	○	○	○	○	×	×	
INPUT$	○	×	○	○	×	×	
LINE INPUT#	○	○	○	○	×	×	
LOAD	○	○	○	○	×	×	
LOC	○	×	○	○	×	×	
LOF	○	×	○	×	×	×	
OPEN	○	○*	○	○*	○*	○	
PRINT# (PRINT#USING WRITE#)	○	○	○	×	○	○	
PUT	○	○	×	×	○	○	
SAVE	○	○	○	×	○	○	
WIDTH	×	×	○	×	×	○	
CHAIN/MERGE	○	×	×	×	×	×	

10.4.1　ディスク装置

ディスク装置は，現在パーソナルコンピュータの外部記憶装置として最も汎用的な装置です．次に示す種々のディスク装置をファイルとして取り扱うことができます．

640KBフロッピィディスク
1MBフロッピィディスク
5MB固定ディスク
10MB固定ディスク
20MB固定ディスク

ディスク装置の取り扱い方に関しては「第11章　ディスクの使い方」および「第12章　ディスクの構造」で詳しく説明しています．

10.4.2　RS-232Cインタフェース

RS-232Cインタフェースで扱うデータも，ファイルという概念で取り扱われています．RS-232Cインタフェースファイルは，ファイルディスクリプタを持ち，デバイス名とファイル名とによって表わされます．

RS-232Cインタフェースのファイルディスクリプタは

> “COMn:〈パリティ〉〈データビット長〉〈ストップビット長〉〈Xパラメータ〉〈Sパラメータ〉”

で表わされます．各パラメータの値は，term文と同じです（詳細は「第13章ターミナルモード」を参照してください）

nには1，2，3のいずれかの値が指定でき，それぞれ第1回線，第2回線，第3回線に対応します．nを省略すると，第1回線に対する処理要求とみなされます．

BASICファイル処理のための基本命令が，RS-232Cインタフェースを扱う入出力処理命令として使用できます．このようなRS-232Cインタフェースの使用方法を「ターミナルモード」に対比させて，「BASICによる入出力モード」とよびます．

入出力モードは，BASICの一般的なプログラミングによって，他のコンピュータとのデータ交換，制御機器や計測機器の制御などの処理を可能にします．

入出力モードを使うときの要点を説明します．

(1) ボーレートと同期クロックの設定

ボーレートの指定はターミナルモードと同じように，メモリスイッチで行います．

同期クロックの設定はディップスイッチSW 1のスイッチ5で行ってください（必ず内部同期指定を選んでください．スイッチはOFFにしておきます）．

外部同期を使用するためには，RS-232Cをシンクロナスモードで使用する必要があります．これは機械語のINまたはOUT命令を使ってRS-232Cインタフェースの初期化，新しい手順，コマンド等を作成することになります．市販のパッケージの中にはこれらを組み込んで外部同期を使用しているものもあります．

(2) BASICで使用できる入出力命令

BLOAD	機械語プログラムを受信します．
BSAVE	機械語プログラムを送信します．
CLOSE	ファイルを閉じます．RS-232Cによる送受信を終了します．
EOF（関数）	バッファが空であると値を真とします．
INPUT#	RS-232Cインタフェースファイルバッファから，データを入力（受信）します． INPUT#文のデータの区切りは文字型では（,）（C_R）（“），非文字型では（,）（△）（C_R）のいずれかです．これらの区切りを受信するまで，データの入力を待ち続けます．
INPUT$（関数）	RS-232Cインタフェースファイルバッファから指定された文字数だけ入力します．
LINE INPUT#	LINE INPUT#文のデータの区切りは（C_R）だけです．（C_R）以外はすべてのデータとして入力します．（C_R）を受信するまで，データの入力を待ち続けます．
LOAD	プログラムを受信します．（プログラムの受信が終了してもLOADコマンドは自動的に終了しません．STOPキーにより強制的に終了させてください）．

LOC（関数）	入力（受信）バッファ中に受信されているデータの文字数を知らせます．
LOF（関数）	バッファの残りバイト数を知らせます．
OPEN	通信の形式を定義し，RS-232Cインタフェースファイルバッファを開設します． ファイル番号，入出力動作モードを定義します．
PRINT#	データを出力（送信）します．式と式の区切りには，(，)か（；）を使います．
PRINT # USING	フォーマット付きで，データを出力（送信）します．
WRITE#	PRINT#文と同様ですが，区切り記号としてコンマを必ず出力し，不要な空白の出力は行いません．
SAVE	プログラムを送信します．
WIDTH	RS-232Cポートに対し，指定されたサイズのデータを送出した後，(C_R)コードを生成，送出します．
WIDTH#	WIDTHと同様に機能します．

(3)　**割り込み処理**

input#文およびline input#文を実行すると，外部からのデータの入力を完了するまで（区切り記号を受信するまで），プログラムは停止しています．この時間に他の仕事を実行させたいときに役立つのが，on com gosub文およびcom on/off/stop文です．これらの文を使用することによりデータが入ってくるまでの間，他の処理を行っており，データを受信すると指定された行番号，もしくはラベルのサブルーチンへ制御を移行するといった動的なデータ受信処理が可能になります．

on com〔(n)〕gosub〈行番号〉または〈ラベル〉

（n）のnには回線番号（1，2，3のいずれか）を指定します．（n）を省略すると第1回線に対する処理要求とみなされます．

（例）
```
5 OPEN "COM : E72XS"AS #1
10 ON COM GOSUB *REC
20 COM ON
        :
1000 *REC
1100 IF LOC(#1)=0 THEN RETURN
1200 PRINT INPUT$(LOC(1), #1) ;
1300 RETURN
```

com on/off/stop

on com gosubで指定した割り込み処理ルーチンへ実際に飛ぶか(割り込みを許可するか),割り込みを禁止するか，一時保留させるかのスイッチです．

（書式）　com〔(n)〕(on / off / stop)

（例）　COM ON（割込み許可）
COM OFF（割込み禁止）
COM STOP（割込み一時保留）

注 意 割り込み処理でデータを受信する場合，INPUT#文は使用しないでください．割り込み処理でINPUT#文を使用しますと，割り込み処理のタイミングがくるい正しくデータを引取れない場合が出てきます．INPUT$文を御使用ください．

(4) **DELコード受信時の処理**

DELコード（$(7F)_{16}$，$(FF)_{16}$）を入出力モードで受信すると，メモリスイッチSW3のビット位置7の状態によって異なったコードとして受信処理を行います．

メモリスイッチSW3のビット位置7が0（OFF）の場合はDELコード（$(7F)_{16}$，$(FF)_{16}$）として処理し，1（ON）の場合はNULコード（$(00)_{16}$）として処理します．

注 意 RS-232Cの第2回線/第3回線を使用する場合，専用のインタフェースボードが必要となります．

10.4.3 プリンタ

プリンタは，一般的にLPRINT/LPRINT USING等の命令で直接制御しますが，ファイルとして扱うことも可能です．

プリンタファイルのデバイス名はLPT：あるいはLPT1：です（どちらを使用してもかまいません）．プリンタファイルは必ずOUTPUTモードでOPENしなければなりません．通常，プリンタファイルに対するデータの出力には，PRINT#，PRINT#USINGあるいはWRITE#文を使用します．

これらの命令によるプリンタファイルへのデータ出力は，ディスクファイルへのシーケンシャルアクセスと同じ考え方でおこないます．

例
```
100 OPEN "LPT : " FOR OUTPUT AS #1
200 PRINT #1, 123, 456
```

10.4.4 キーボードとスクリーン

キーボードとディスプレイは一般的にINPUT文/PRINT文等で直接操作しますが，ファイルとして扱うこともできます．この機能は，フロッピィディスクやプリンタを扱うプログラムを作成する場合に利用すると非常に便利です．

例えば，フロッピィディスク上のファイルからデータを読み込んで処理を行い，再びフロッピィディスクの別のファイルに書き込むプログラムを作成するとします．この場合に，最初からフロッピィディスクを用いてプログラムを作成するよりも，いったんキーボードからデータを入力し，スクリーンへデータを表示させれば，プログラムのデバッグが簡単におこなえます．そこで，プログラムを作成している間は，次のようにファイルをオープンします（キーボードは必ずINPUTモードでOPENしなければなりません．また，スクリーンは必ずOUTPUTモードでOPENしなければなりません）．

```
 100 open "kybd : test1" for input as # 1
 200 open "scrn : test2" for output as # 2
  :
1000 input # 1, a$
  :
2000 print # 2, b$
```

⋮

プログラムが完成すれば，行番号100と200の2行を変更すればフロッピィディスク用のプログラムになります．（行番号1000や2000の入出力文は変更しなくても良いわけです．）

```
100 open "1 : test1" for input as # 1
200 open "2 : test2" for output as # 2
        ⋮
```

注　意　キーボード（KYBD：）およびスクリーン（SCRN：）では，使用できない入出力命令がありますから注意してください．また，"SCRN："を使用し，スクリーンに対しPRINT #文で日本語を表示することはできません．

スクリーンに日本語が表示できるのはPRINT文のみです．

10.4.5　カセットテープ

本体の電源を切ったり，リセットスイッチを押したりすると，プログラムやデータは，すべて消えてしまいます．ですから，再び使用する可能性のあるプログラムやデータは，フロッピィディスクやカセットテープに保存しておきます．メモリの中のプログラムやデータをフロッピィディスクやカセットテープに書き込むことを，「**セーブ**」するといい，逆にプログラムやデータをフロッピィディスクやカセットテープからメモリへ読み込むことを「**ロード**」するといいます．

(1)　**プログラムのロード，ベリファイとセーブ**

カセットテープへのロードとセーブは，どのBASICモードでも実行できますが，ここでは，ROM BASICモードの場合を説明します．使用するカセットテープレコーダーは，一般用オーディオカセットテープレコーダーです（以後，テープレコーダとよびます．）

(a)　テープレコーダの接続

次の点に注意してください．

①　リモート端子（黒色）を接続した場合

リモート端子を接続しますと，テープレコーダのモータのON/OFFは，BASICがコントロールします．ですから，PLAYボタンを押していても，カセットテープを使用する命令が実行されなければ，カセットテープは回りません．この機能によって，必要な間だけ，カセットテープを回すことができ，カセットテープの無駄使いが防げ，また，手間もかかりません．

②　リモート端子を接続しなかった場合

この場合は，テープレコーダのスタート，ストップは，すべて自分でやる必要があります．ですから，カセットテープを使用する命令を実行する前にPLAYボタンを押し，終るとSTOPボタンを押すという操作が必要になります．

普通，1つのプログラムをロードしたり，セーブする場合は，リモート端子を接続せずに操作することが多いようですが，プログラムの中でデータのロード，セーブを行う場合は，リモート端子を接続していないと，うまく動作しない場合があります．以後，リモート端子が接続されているとして説明をします．リモート端子を接続しない場合は，先に「(4)　リモート機能を使用しない場合」を読んでください．

(b)　プログラムのセーブ

テープレコーダが接続できたら，キーボードから，次のプログラムを入力してください（入力が終っても実行させないでください）．

```
new ↵
100 input a,b,c,d ↵
200 print "Set tape, then push PLAY and REC" ↵
300 input "OK" ; a $ ↵
400 open "cas : suchi" for output as #1 ↵
500 print #1,a,b ↵
600 print #1,c,d ↵
700 close #1 ↵
800 end ↵
```

入力したプログラムがまちがっていないかどうか，ディスプレイとよく見比べてください．まちがっていなければ，このプログラムをカセットテープへ次の順序でセーブしてください．

1．記録してもよいカセットテープを，テープレコーダにセットします．プログラムやデータを入力すると，当然，前に記録してあったものは消えてしまいます．また，このとき，テープカウンタを0にセットしておけば，どこからセーブしたのかすぐりわかります．

2．テープがセットできたら，録音ボタン（REC）とPLAYボタンを押します．このとき，リモート端子（黒）が接続してあれば，カセットテープは回り始めないはずです．そこで命令を入力します．

```
motor ↵
```

↵ を入力すると同時に，"カチッ"という音がして同時にテープが回り始めます．数秒たったらもう一度，

```
motor ↵
```

を入力します．すると再び"カチッ"という音がして，テープは止ります．これでセーブの準備が整ったわけです．

注　意　カセットテープには，始めと終りにクリアフィーダという磁性体が塗ってない部分があります．この部分には記録できませんから，プログラムやデータをセーブする前に空送りしておく必要があります．また，前のプログラムやデータとの区別をつけるためにも，セーブする前は，motor文でテープを送る習慣をつけてください．

NOTE　motor文は，モーターをコントロールします．

motor ↵ ⇨モータースタート⇨motor ↵ ⇨モーターストップ

3．いよいよプログラムをセーブします．PLAYボタンと録音ボタンは押したままです．そこで，次の命令をキーボードから入力します．

```
save "cas : test" ↵
```

4．モーターが回り始め，しばらくすると"OK"が表示され，モーターがストップしてプログラムのセーブは終ります．

ここで使用したsave文が，プログラムを「セーブ」する命令です．

save “〈ファイルディスクリプタ〉” [↵]

今の場合ですと，ファイルディスクリプタは“cas : test”です.「cas:」がデバイス名で「t-est」がファイル名です.

注 意　デバイス名「cas：」

カセットテープのデバイス名には「cas：（又は，cas1：)」と「cas2：」の2つがあります．この2つのデバイス名は，テープレコーダーとのデータ転送速度を指定するために準備されています．詳しくは，「(3)　カセットテープの転送速度」を参照してください．

NOTE　**カセットテープへのプログラムのセーブはsave文で行います.**

save “cas : XXXX”（1200ボー）

又は，

save “cas2 : XXXX”（600ボー）

(c)　ベリファイ

カセットテープにプログラムをセーブしたときに，正しくセーブされたかどうかチェックする機能があります．LOAD？命令です．この命令を使って，メモリの中のプログラムとカセットテープにセーブされたプログラムが等しいかどうかを比べます．この操作を“ベリファイ”といいます．ではベリファイの実行の仕方を説明します．

1．プログラムのセーブを始めたところまで，テープを巻き戻します.
（家庭用テープレコーダーを使用し，かつリモート端子を接続している場合は，まずmotor文を使用し，つぎに巻き戻しを行い，終ったら，モータストップ（motor [↵]）を実行しておきます).

2．テープレコーダの音量と音質を調節します．音量は，普段使用する場合よりもやや大きめにセットし，音質が調節できる場合は，高音が強調される側へセットします.

3．PLAYボタンを押した後で，次の命令を実行します.

load? “cas : test” [↵]

4．“test”というプログラムが見つかると

Found : test

と表示し，読み込んだデータとメモリの内容をチェックします.

5．カセットテープにセーブされている内容が一致した場合は，

Ok

と表示し，一致しない場合は，

Bad

と表示してきます.

一致しなかった原因としては，次の場合が考えられます.

1．再生用の音量と音質の調整が適当でなかった.
この場合は，音量と音質を調整し直して，何度か繰り返してみてください．使用しているテープレコーダによって，音量が，ある特定の範囲でしか動作しないものがあります.

2．セーブがうまくできない場合

音量，音質をいくら調整してもBadと表示される場合は，セーブした時に，失敗したことが考えられます．その原因としては，次のようなことが考えられます．

① ケーブルの接続が正しくなかったり，接続プラグが，しっかりと差し込まれていなかった．

② 録音音量が調整できるテープレコーダを使用したが，その調整が正しくなかった．

③ セーブするときに，カセットテープのクリアフィーダの部分をスキップするのを忘れていた．

それぞれ，もう一度よく確認してから，セーブを行ってください．

(NOTE) プログラムをカセットテープにセーブした場合は，必ずベリファイを実行して，確認してください．

(d) ロード

今度は，カセットテープにセーブしたプログラムを，再びメモリへロードしてみます．

まず，メモリにプログラムが残っている状態からはじめます．

[f·4] [↵] （または，list [↵]）

を入力すれば，メモリに入っているプログラムが表示されます．

次にメモリに入っているプログラムを消します．

new [↵]

この命令を実行すると，あなたの入力したプログラムは，すべて消えてしまいます．ですから，

[f·4] [↵] （またはlist [↵]）

を入力しても，プログラムは表示されません．

ロードを始めます．

1．カセットテープを巻き戻します（家庭用テープレコーダを使用しているとき，リモート端子を接続している場合は，motor文を入力しないとモータは回りません）．

2．音量と音質を確認します（ベリファイしたときと同じです）．

3．PLAYボタンを押します（テープは回りません）．

4．次の命令を入力します．

load "cas : test" [↵]

5．[↵]を入力するとカセットテープが回り始めます．プログラムが見つかると（今の場合は「test」です．）

Found : test

とディスプレイに表示し，ロードしている間は，画面の右上スミに"♥"を表示します．指定したプログラム名以外のプログラムが見つかると，

Skip : XXXX

と，プログラム名をXXXXで表示し，次のプログラムを読みに行きます．

6．ロードが無事に終了すれば，

Ok

が表示され，カセットテープは止まります．

もし，ロードが途中で失敗すれば，ディスプレイに，

Tape read errorというメッセージが表示され，ブザー（Beep）音が鳴ります．エラーとなった場合には，ベリファイのところで説明した原因を１つ１つチェックしてみてください．何度，ロードしてもエラーとなる場合は，テープにキズがある場合などが考えられます．また，ベリファイを行っていなければ，うまくセーブができていない可能性があります．このような場合は，もう一度，セーブからやり直す必要があります．

ロードがうまく行けば，

[f･4] [↵] （またはlist [↵]）

でプログラムが表示されます．

注意 ロードがうまくいかなかった場合に，

[f･4] [↵] （またはlist [↵]）

を実行すると，途中までのプログラムが表示される場合がありますが，このプログラムを実行させたり，新たな行をつけ加えて実行させることはできません．new文を実行して，最初から入力してください．

(2) データのセーブ，ロード

次は，データのセーブ，ロードの方法を，前節で使ったプログラムを例にとって説明します．

(a) データのセーブ

1．プログラムのセーブとロードに用いたプログラムをロードします．

2．このプログラムを実行させます．

[f･5] （または，run [↵]）

3．すると画面に，“？”を表示します．これは，行番号100の input文が，キーボードからの入力を要求しているのです．(input文については，「BASICリファレンスマニュアル」を参照してください．)

1，2，3，4 [↵]

と入力します．

4．すると今度は，次のメッセージが表示されます．

Set tape, then push PLAY and REC
OK?

5．そこで，データを入力してもよいカセットテープをテープレコーダにセットします．このとき，テープカウンタを０にしておけば，後でどこからデータが入っているか一目でわかります．

6．カセットテープのセットが終ったら，テープレコーダの録音ボタンとPLAYボタンを押します．これらの準備が終ったら，

[↵]

を入力します．

7．[↵]を入力すると，すぐに，カセットテープが回りはじめ，しばらくして，ディスプレイに“Ok”と出力し，カセットテープは止まります．

これで，キーボードから入力した数値が，カセットテープにセーブされたわけです．

このプログラムを簡単に説明します.

```
100 input a,b,c,d
200 print "Set tape, then push PLAY and REC"
300 input "OK" ; a $
400 open "cas : suchi" for output as #1
500 print #1,a,b
600 print #1,c,d
700 close #1
800 end
```

① キーボードから数値を入力し，カセットテープのセットも終り，録音ボタンとPLAYを押して，リターンキー（⏎）を入力すると，プログラムは次に，ファイルのオープンを行います.（行番号400）

② open文が実行されると，カセットテープへのデータを出力するための窓口（バッファ）が割り当てられ，データのセーブのための準備が整います.

③ printを用いてデータをセーブします.（行番号500，600）

print文は，ディスプレイに文字を表示させるだけではなく，このように〈ファイル番号〉を指定すると，ファイルへのデータの出力を行います.

④ データのセーブが終ったら,close文でファイルを閉じておきます.（行番号700）

(b) データのロード

(a)でセーブしたデータを，ロードしてみましょう.

1．次のプログラムを入力します.

```
100 a=0 : b=0 : c=0 : d=0 ⏎
200 print "push PLAY" ⏎
300 open "cas : suchi" for input as #1 ⏎
400 input #1,a,b ⏎
500 input #1,c,d ⏎
600 print a,b,c,d ⏎
700 close #1 ⏎
800 end ⏎
```

このプログラムの意味は，おおよそ理解していただけると思います．データのセーブのプログラムとの主な違いは，ファイルをオープンするときに，データの入力用として"input"を宣言していることと，データのロードを"input #1"で行っていることです.

2．プログラムが入力できたら，データをセーブし始めたところまで，カセットテープを巻き戻します.

3．カセットテープの準備ができたら，実行させます.

f·5 （または，run ⏎）

4．"push PLAY"というメッセージが現われたら，PLAYボタンを押します．カセットテープが回りはじめて，しばらくすると，ロードされたデータがディスプレイに表示されます.

(c) データの型と数の一致

カセットテープにデータをセーブ，ロードする場合の注意事項として，セーブするときとロードするときの〈データの型と数の一致〉があります．データとして文字をセーブしたならば，それを数値データとしてロードすることはできません．また，１つのprint文で４つのデータをセーブしている場合には，input文では２つしかロードしないというわけにもいきません．ただし，データの型と数が一致していれば，ロードしてきたデータを代入する変数は，セーブしたときに使った変数と異なっていてもかまいません．

〔良い例１〕

セーブするとき，

```
10 open "cas : sample" for output as #1
        :
100 print #1,a,b,c
        :
500 close #1
```

ロードするとき

```
10 open "cas : sample" for input as #1
150 input #1,X,Y,Z
        :
300 close #1
```

データの型（数値），数（３つずつ）が一致しているのでOKです．

〔良い例２〕

セーブするとき

```
10 open "cas : sample" for output as #1
        :
100 print #1, yama $, kawa $
        :
500 close #1
```

ロードするとき

```
10 open "cas : sample" for input as #1
        :
200 input #1, tani $, umi$
        :
600 close #1
```

これも，データ型（文字），数（２つずつ）が一致しているのでOKです．

〔悪い例〕

セーブするとき

```
10 open "cas : sample" for output as #1
        :
100 print #1,a,b,c,d
        :
700  close #1
```

ロードするとき

```
10 open "cas : sample" for input as #1
        :
200 input #1,a,b
210 input #1,c,d
        :
800 close #1
```

データの型（数値）は一致していますが，１つのprint文で４つのデータをセーブしたのに対し，ロードするときは，１度に２個しか扱っていません．これでは，ロードできません．

NOTE データのセーブとロードには，データの型と数とが一致することが条件です．

(3) カセットテープの転送速度

カセットテープを使用する場合に，「600ボー」と「1200ボー」の２つの転送速度を使用します．

この指定は，デバイス名で区別します．

デバイス名	転送速度
cas : (cas 1 :)	1200ボー
cas 2 :	600ボー

同じ長さのプログラムやデータをセーブしたりロードしたりする場合，1200ボーですと，600ボーの半分の時間で実行できます．ところで，600ボーでセーブしたプログラムやデータを1200ボーでロードすることはできません．また，逆に1200ボーでセーブしたプログラムやデータを600ボーでロードすることもできません．ですから，プログラムやデータをセーブする場合には，どちらの転送速度でセーブしたのかを記録しておいて下さい．

注 意 同じテープレコーダで，セーブ，ロードを行う場合は，「1200ボー」で行う方が速く実行できますが，セーブしたテープレコーダと，ロードするテープレコーダが異な

る場合に，「1200ボー」ですと，モーターの回転速度の誤差などから，エラーが発生する恐れがあります．このようなことが考えられる場合は，「600ボー」でセーブ，ロードを行ってください．

(4)　リモート機能を使用しない場合

テープレコーダを接続する場合，リモート端子を接続しておくと，テープレコーダのモータの「ON」「OFF」はシステムがコントロールします．しかし，リモート端子が接続できない場合は，自分でコントロールしなければなりません．

1．セーブするとき

PLAYボタンと録音ボタンを押した後，セーブを行う命令を実行します．

2．ロードするとき

ロードを行う命令を実行したあと，PLAYボタンを押します．

ただし，データのロード，セーブでは，リモート機能を使用しないと実行できない場合がありますから注意してください．

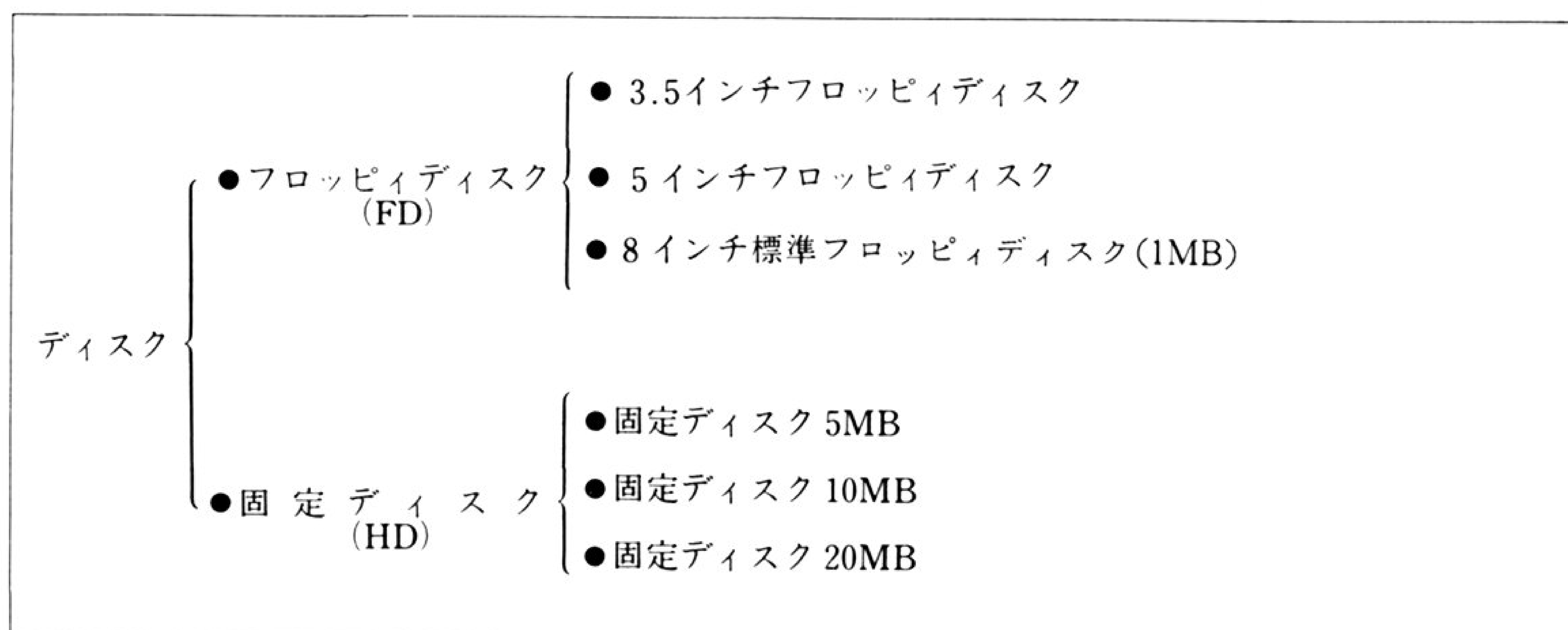
注　意　**カセットテープを用いて，データのセーブ，ロードを行う場合は，リモート端子を接続して使用してください．**

第11章

ディスクの使い方

最初に，ディスク媒体，ディスク装置（ユニット）についての名称，内容について整理しておきます．

(1) 使用できる「ディスク」の種類と名称



(2) 使用できるフロッピィディスク媒体とその略称

媒体の種類＼媒体の型		3.5インチフロッピィディスク	5インチフロッピィディスク	8インチフロッピィディスク
両面	倍密度	／	／	8″ 2D (容量：1MB)
	倍密度倍トラック	3.5″2DDあるいは640KB フロッピィディスク (容量：640KB)	5″ 2DDあるいは640KB フロッピィディスク (容量：640KB)	／
	高密度	3.5″2HDあるいは1MB フロッピィディスク (容量：1MB)	5″ 2HDあるいは1MB フロッピィディスク (容量：1MB)	／

表の見方 1．［／］の項は定義されていないか，使用できない媒体です．

2．倍密度とは1セクタ256バイトからなる媒体を示します．

3．8インチフロッピィディスクの倍密度の媒体では，サーフェス番号0，トラック番号0だけは1セクタ128バイトでフォーマットされ，他は256バイトでフォーマットさ

れています．

4．3.5” 2DDフロッピィディスクと5” 2DDフロッピィディスクは同一の論理構造を持ちます．

5．3.5インチ2HD，5インチ2HDおよび8インチ2Dフロッピィディスクは同一の論理構造を持ちます．

11.1　ディスクの使用準備

11.1.1　フォーマッティング

購入したばかりの未使用のディスクは，そのままでは使用することができません．ディスクをフォーマッティングする必要があります．ユーティリティプログラム“format.nip”を使ってフォーマッティングしてください．

フォーマッティングのレベル ／ 未使用ディスクの種類	第1レベルのフォーマッティング（イニシャライズ）（注1）	第2レベルのフォーマッティング（システム　フォーマッティング）
3.5インチおよび5インチフロッピィディスク	行う必要あり	行う必要あり
8インチ標準フロッピィディスク	行う必要なし	行う必要あり
固定ディスク	行う必要あり	行う必要あり

（注1）物理フォーマッティングともいいます．

11.1.2　固定ディスクユニット固有機能の準備

固定ディスクユニットを効率よく利用するために固定ディスクに固有な機能が用意されています．これらの機能を使用するには，そのための準備が必要です．ここでは固有な機能とそのための準備の方法について説明します．

(1)　3つの固有な機能

①　ユーザ識別名を使用して固定ディスクユニットを論理的な複数の独立した媒体として使用する機能

（マルチユーザ機能）

②　デバイス名指定を固定ディスクから行う機能

（デバイス名優先指定機能）

③　複数のシステムによってディスク領域を分割使用する機能

（領域分割機能）

(2)　固有な機能の使用準備

固定ディスクユニット上のファイルには利用者が指定するファイル名に加えて，ユーザ識別名を

項番	固有な機能	使用のための設定方法			主な機能および指定方法
1	マルチユーザ機能	メモリスイッチSW 5のビット2	0	OFF	システム立ち上げ時の画面表示で次の問い合わせが表示される． User identifier ? これに応答した文字列がユーザ識別名として使用される
			1	ON	システム立ち上げ時User identifier ?の問い合わせはおこなわれない．固定ディスク内のすべてのファイルは「共通ユーザ識別名」で管理されます
2	デバイス名優先指定機能	メモリスイッチSW 5のビット1	0	OFF	デバイス名の指定順序（「11.2」参照）フロッピィディスク→HDの順となる
			1	ON	デバイス名の指定順序（「11.2」参照）HD→フロッピィディスク
3	領域分割機能	ユーティリティプログラム format.hd を使用し， format sizeを指定する			フォーマットするサイズは（メガバイト）？の問い合わせに対して，BASICシステムで使用するディスク領域を指定します．他の領域はCP/M-86，MS-DOS等で使用します（「15.6」参照）

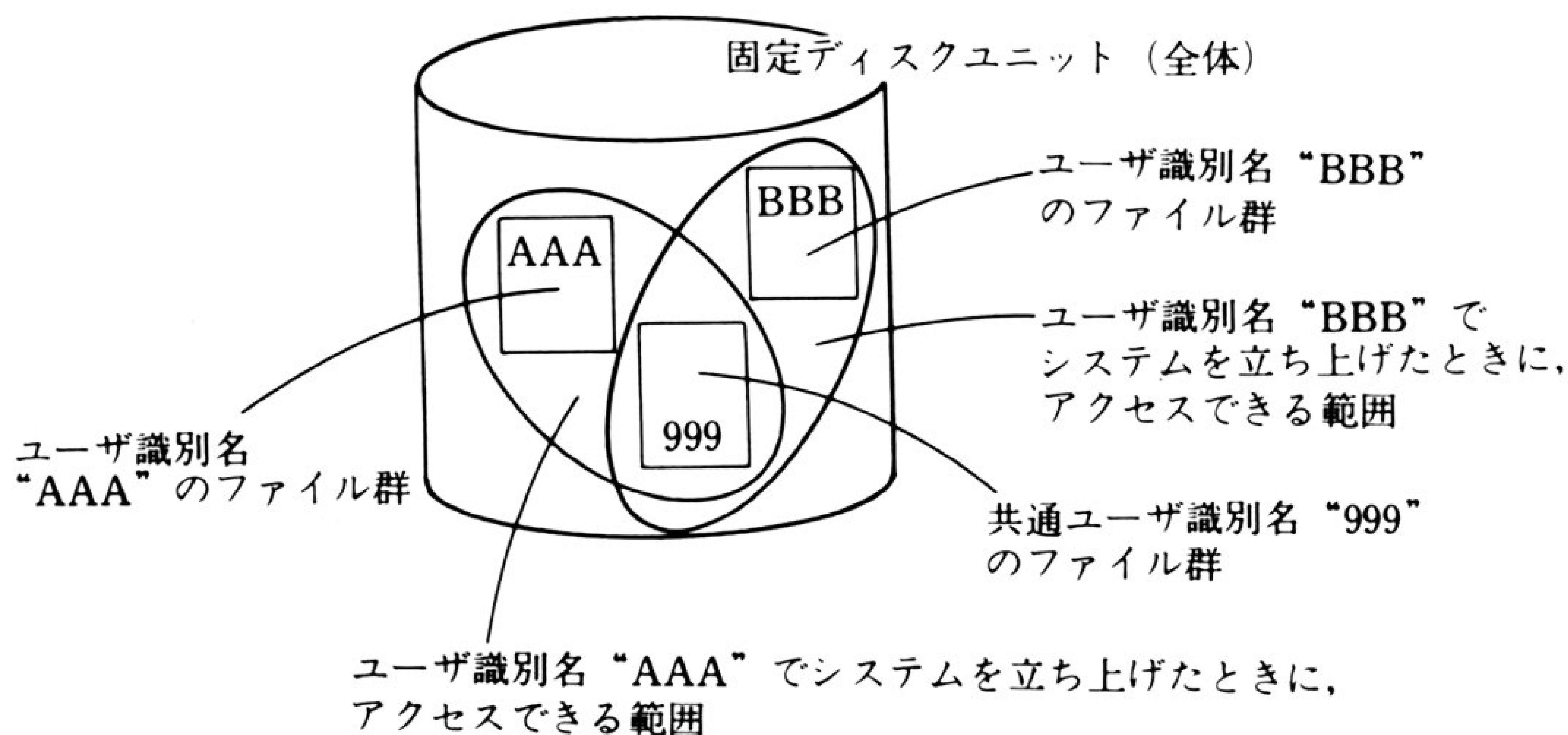
拡張ファイル名として追加しています．このユーザ識別名によって固定ディスク上のファイル群を論理的に区別しています．この区別がシステム運用中にアクセスできる対象ファイルの範囲を定義します．

使用できる対象のファイル群を，システム立ち上げ時の操作で指定します．これが User identifier ?に対する応答です．この応答で3桁までの文字列を入力します．これが「ユーザ識別名」です．

指定したユーザ識別名は，そのシステムが立ち上っている間中，ファイルを作成する場合には，拡張ファイル名としてシステムが自動的にディレクトリに書き込み，以後の参照のために利用します．また，ファイルを参照する場合には，システム立ち上げ時に指定したユーザ識別名をもったファイルだけを対象に参照することができます．ただし，共通ユーザ識別名“999”をもったファイルは常に参照することができます．なお，実際のファイル作成，ファイル参照等において指定するファイル名は今までと同じように，ファイル名（正しくは，ファイルディスクリプタ）だけを指定します．

また，1つのユーザ識別名のもとで定義できるファイル名は，共通ユーザ識別名“999”の下で

作成されているファイル名と，重複することができません．重複したファイルを定義しようとするとエラーメッセージFile write protectedが表示されます．



ユーザ識別名“AAA”からアクセスできるファイル

ファイルアクセスモード ＼ 対象ファイル名			“AAA”の下のファイル（同じユーザ識別名）	“BBB”の下のファイル（他のユーザ識別名）	“999”の下のファイル（システム共通識別名）
OPEN “～” FOR APPEND		更新	○	×	○
OPEN “～” FOR INPUT		参照	○	×	○
OPEN “～” FOR OUTPUT		作成	○	×	×
OPEN “～”（ランダムアクセスの場合）	PUT	更新	○	×	○
	GET	参照	○	×	○
	PUT	作成	○	×	×
KILL		削除	○	×	×

表の見方 1．“～”はファイルディスクリプタを示す．

2．この表はユーザ識別名“AAA”で立ち上げたシステムのもとで，どのファイルならアクセスできるかを示しています．

3．ユーザ識別名“999”をもつファイルの作成・削除はユーザ識別名“999”で立ち上げたシステムでないとできません．

注　意　ユーザ識別名“999”をもつファイルは“sys.～”，“～．sys”のように，一意的なファイル名になるように定義することをおすすめします．

一般のユーザ識別名でファイルを作成する場合に，システム共通識別名をもつファイルと重複するファイル名でファイルを作成しようとするとエラーになります．また，誤って削除しないように予防する必要があります．これらのためにも上記のようなファイル名を使用することをすすめます．

11.2　ディスクにおけるファイルディスクリプタ

ディスクのファイルディスクリプタは，ドライブ番号とファイル名で構成されます．

〔ファイルディスクリプタ〕＝〔ドライブ番号〕＋“：”＋〔ファイル名〕

例　“1：SOURCE.BAS”“3：M1”

ディスク装置については，ドライブ単位にドライブ番号が定義されており，最大14ドライブまで定義することができます．

(1)　ディスクユニットとドライブの関係

ユニットは装置全体を示し，ドライブはユニットの内にあります．ディスク媒体を装着し，動作させる機構です．たとえば，8インチ標準フロッピィディスクユニットPC-9881Nには2つのドライブがあります．

(2)　ドライブ番号

ドライブ番号は各ドライブに割振られた一連番号です．

(3)　フロッピィディスクユニットのドライブ番号

フロッピィディスクユニットに対するドライブ番号の割り当ては，標準のシステムディスクを使用する場合，一般に次のようになります．

・本体にディスクユニットが内蔵されている場合

内蔵のディスクユニット＃1→内蔵のディスクユニット＃2→外付けの拡張ユニット（システムディスクと同タイプのディスクユニットが優先します）．

・本体にディスクユニットが内蔵されていない場合

システムディスクと同タイプの外付けのユニット＃1→システムディスクと同タイプの外付けのユニット＃2→外付けの拡張ユニット（システムディスクと同タイプのディスクユニットが優先します）

このルールはシステムディスクの標準として定められているもので，利用者は必要に応じて，フロッピィディスクユニットに対してドライブ番号の割振り方を変更することが可能です．1MBタイプと640KBタイプのどちらのユニットを優先してドライブ番号を割振るかの指定ができます．

システムディスクのフォーマッティングやDISK codeのコピー時に，その指定が可能です．詳細は，「第15章システムディスクに格納されているユーティリティプログラム」のformat.nipあるいはsysgen.nipの節を参照して下さい．

また，本体にディスクユニットが内蔵されており，さらに同タイプの拡張ユニットが接続されている場合，内蔵のディスクユニットと拡張ユニットのどちらを優先してドライブ番号を割振るかを指定することができます．標準では内蔵のディスクユニットを優先してドライブ番号を割振るようになっていますが，DIP SW1の4番スイッチをONにしますと，拡張ユニットを優先してドライブ番号を割振ります．

(4)　固定ディスクが接続されている場合のドライブ番号

固定ディスクが接続されている場合，固定ディスクを効率よく使用するために2通りのドライブ番号指定方式が用意されています．

(i)　標準指定方式（メモリスイッチSW5・2^1ビットが0の場合）

フロッピィディスクユニットを優先してドライブ番号を割当てます．

(ii) 固定ディスク優先指定方式（メモリスイッチSW5・2^2ビットが1の場合）

HDを優先してドライブ番号を割当てます.

HD→フロッピィディスクの順で1～14までのドライブ番号が割振られます.

NOTE 固定ディスク優先指定方式の利点

① 固定ディスクをベースにしたシステムとして運用することができる.

② 固定ディスク上のファイルを使用する場合に，デバイス名を無指定（デフォルト）で使用できる.

標準指定方式の利点

① フロッピィディスクをベースにしたシステムとして運用することができる.

② 特定の大量データファイルのみを固定ディスクにおいて使用することができる.

11.3 プログラムのセーブ・ロード

11.3.1 プログラムのセーブ

ディスクにプログラムをセーブする場合は，save文を使います.

```
save 〈ファイルディスクリプタ〉〔, a/p〕
```

（例） save "prog 1"

save "prog 1.n88"

save "2: prog 1"

save "test 1", a

save "test 2", p

通常のプログラムのセーブ以外に，Aオプションと Pオプションを付けたセーブができます．このAオプションとPオプションについては，「11.3.3 ファイル名のリストをとる」で説明します．この他に，機械語をディスクにセーブすることもできます．この機械語のセーブにはbsave文を使います.

```
bsave 〈ファイルディスクリプタ〉,〈開始番地〉,〈プログラムのバイト数〉
```

（例） def seg = &h1000

bsave "mprog", &ha000, &h20

これは，1a000H（16進数）番地から1a01fH番地の機械語（長さ20（16進数））を"mprog"というファイル名でディスクにセーブする例です.

「11.2 ディスクにおけるファイルディスクリプタ」のところでも述べましたがsave文およびbsave文では，ドライブ番号の指定に気をつけてください．ドライブ番号1にあるファイル名と同じファイル名でドライブ番号2にセーブしようとしてドライブ番号を忘れると，ドライブ番号1のそのファイル名のプログラムが消えてしまいます.

ドライブ番号2に"nec"というプログラムをセーブする場合

（例） × save "nec"

○save "2 : nec"

このことは，後述のファイル削除（kill）においても重要なことです．

NOTE **同じファイル名にセーブしようとした場合は，エラーにならずにそのファイル名のプログラムを消して上書きします．**

11.3.2 プログラムのロード

ディスクにセーブされているBASICのプログラムのロードは，load文によって行います．

load〈ファイルディスクリプタ〉〔, r〕

（例）　load "prog1"

　　load "prog.n88"

　　load "2 : prog1"

　　load "prog1", r

上の例の4番目は，Rオプション付きのload命令で，プログラムをロードした後で，そのプログラムを実行します．詳しくは，「11.3.8Rオプションとrun命令」を参照して下さい．

bsave文のセーブした機械語のロードには，bload文を使います．

bload〈ファイルディスクリプタ〉〔,〔ロードアドレス〕〔, r〕〕

（例）・bload "mprog"

　・def seg＝&h1000

　　bload "mprog", &h000

　・bload "mprog" ,,r

1番目の例は，"mprog"というファイル名の機械語をbsave文でディスクにセーブする際に指定した開始番地からロードします．2番目の例は，ロードする際，&h10000番地から機械語をロードします．最後の例は，1番目の例と同様にプログラムをロードしたあとに，bsave文で指定した開始番地またはロードアドレスからプログラムをスタートさせます．

（「11.3.8　Rオプションとrun命令」を参照してください.）

　データファイルをロードしたら？

このことは規定されていません．最悪の場合は，リセットをかけてもう一度始めから，システムをスタートさせることになります．

この種のトラブルを避けるには，ファイル名でデータかプログラムを区別させるしかありません．その場合はファイルの拡張子をうまく利用してください（「10.1.2　ファイル名」参照)．

（例）　プログラムファイルには必ず".n88"をつける．または，省略する．

　　データファイルには".dat"をつける等により区別をつけます．

11.3.3 ファイル名のリストをとる

ディスクにセーブされているファイルの名前を知りたいときには，次の命令を使います．

files〈ドライブ番号〉

ドライブ番号とは，ファイルディスクリプタのところで述べましたが，ユニットの番号です．

（省略するとドライブ番号は1です.）

（例）　files

```
progra.m        1
ok
```

上の例で，programのaとm（ファイル名の6文字目と7文字目）の間に小数点が表示されていますが，これは，programというファイルが非アスキー（バイナリー）形式でディスクにセーブされていることを意味しています．

プログラムをディスクにセーブするときは，非アスキー形式以外にアスキー形式を指定することができます．

NOTE **プログラムはコード化されてメモリに格納されています．非アスキー形式のセーブはこの形式のままディスクへのセーブをおこないます．一方，アスキー形式のセーブはコード化されたプログラムを入力イメージにもどしてからディスクへのセーブをおこないます．**

これら2つの形式では，当然，非アスキー形式でセーブしたプログラムの方が容量が小さくてすみ，1枚のディスクに，より多くのプログラムをセーブすることができます．また，ロード・セーブに要する時間も短くなります．しかし，アスキー形式でセーブしておかなければできないこともあるので使いわける必要があります．（プログラムのマージとチェインです．11.3.4と11.3.9参照）．

データファイルはすべてアスキー形式です．先程の例で，非アスキーセーブ（通常のセーブ）のファイル名の6文字目と7文字目の間に小数点が書かれていましたが，アスキー形式でセーブされたファイルの場合は，小数点のところが常に空白（スペース）になります．

アスキー形式でプログラムをセーブするときは，次のように，Aオプションを指定します．

save 〈ファイルディスクリプタ〉, a

（例） save "program1", a　　アスキー形式

save "program2"　　非アスキー形式

files

progra m1　　1　　アスキー形式

progra.m2　　1　　非アスキー形式

この例で，ファイル名の後に数字の1が出ていますが，これは，クラスタと呼ばれるファイルの大きさを示す数です（「第12章　ディスクの構造」で述べます）．

ファイル名の6文字目と7文字目の間の小数点または空白（スペース）は，files命令を実行したとき表示されるだけで，ロード（指定）するときは，そのファイルを作ったときにつけたファイル名を用います．

（例） 先の例でセーブしたプログラムをロードするとき

load "program1"

load "program2"

（アスキー形式のファイルをロードする場合でも，",a"は必要ありません．）

「11.3.1　プログラムのセーブ」でふれたPオプションですが，これを指定すると，そのプログラムは暗号化された非アスキー形式でセーブされます．一度Pオプションを指定してセーブすると，そのプログラムの内容を見たり，変更したりできなくなります．このPオプションの指定を解除する方法は準備されていません．Pオプションはソースプログラムの機密保持のために使用します．

(NOTE) Pオプションを指定してセーブしたプログラムは，その内容を見たり，変更したりすることができません．

（例）
```
save "2 : program1", p
files
progra.m1          1      非アスキー形式
```

(NOTE) 機械語のファイル（bsave文でセーブされたファイル）は，6文字目と7文字目の間に"*"が表示されます．

```
def seg = &h1000
bsave "2 : mprog1", &h0000, &h20
files
mprog1  *        1
```

11.3.4　プログラムのマージ

メモリ上にあるプログラムと，指定されたディスク上のファイル（アスキー形式で書かれたプログラム）を行単位（行番号ごと）で合成して，1つのプログラムにすることを「マージする」と言います．

マージする場合，同じ行番号のプログラムがあれば，ディスク上のファイルのその行番号のプログラムが優先して，メモリ上に置き換わります．マージできるファイルは「11.3.3ファイル名のリストをとる」(files命令）で述べたアスキー形式でセーブされたファイルだけです．

merge 〈ファイルディスクリプタ〉

（例）　merge "program1"

program1はアスキー形式でセーブされたファイル．

(NOTE) マージできるプログラムは，アスキー形式でセーブされたプログラムだけです．

11.3.5　ファイルの削除

ディスク上のファイルはkill命令で削除します．

kill 〈ファイルディスクリプタ〉

（例）
```
kill "prog1"
kill "2 : prog2.n88"
```

注　意　「11.2　ディスクにおけるファイルディスクリプタ」のところでも触れましたが,ドライブ番号の指定には充分注意してください．一度killしたプログラムはもとにはもどりません．

11.3.6　ファイルの属性とset文

ファイルの属性には，次の2種類が指定できます．

(1)　書き込み禁止

(2)　read after write

（ファイルにデータを書き込む際，いったん書いたデータをもう一度読み直して正しく書けているかチェックしていきます.）

通常，何も指定しなければ，どちらの属性も持っていません．特に重要なファイルであるならば，(1)の書き込み禁止属性を指定しておきます．ファイルの内容の変更が必要になった時には，そ

の属性を解除して，ファイルの内容を変更してください．必要であれば，もう一度，書き込み禁止属性を指定しておきます．

属性の指定，解除にはset文を使います．

set 〈ファイルディスクリプタ〉,〈属性文字〉

属性文字はp, P, r, R〈それ以外〉であり，それぞれ

p, P……書き込み禁止

r, R……read after write

それ以外……属性の解除

（例） set “prog1”, “P” 書き込み禁止

set “prog1”, “ ” ファイル名prog1のすべての属性の解除

このファイルの属性は，ディスクに保存されます．

次に，他の形式のset文を紹介します

(a) set〈ドライブ番号〉,〈属性文字〉

現在そのドライブに入っているディスクの属性を指定します．以後，このディスクに作られるファイルは，その属性を持ちます．

（例） set 1, “R”（read after write）

set 2, “P”（書き込み禁止）

(b) set #〈ファイル番号〉,〈属性文字〉

この文は，指定したファイル番号のファイルの属性を指定しますが，ファイルがオープンされている間だけ有効です．

（例） set #1, “R”（read after write）

11.3.7 ファイル名の付けかえ

name命令によって，ディスクに書き込まれているファイル名を変更することができます．

name〈旧ファイルディスクリプタ〉as〈新ファイルディスクリプタ〉

2つのファイルディスクリプタのドライブ番号は，同じでなければなりません．

（例） name “DEMO” as “SOFT” ○

ファイル名“DEMO”→ファイル名“SOFT”

name “2 : TEST” as “2 : WORK” ○

ファイル名“TEST”→ファイル名“WORK”

name “2 : WORK” as “WORK1” ×

11.3.8 Rオプションとrun命令

メモリ上のBASICのプログラムを実行させるrun命令以外に，ディスク用のrun命令があります．

run〈ファイルディスクリプタ〉

この形式のrun命令は，ファイルディスクリプタで指定されるプログラムをロードして，通常のrun命令を実行します．

（例） run “test”→load “test”

run

Rオプションはload, bload, run命令に付加することができます.

```
load 〈ファイルディスクリプタ〉, r
bload 〈ファイルディスクリプタ〉, r
run 〈ファイルディスクリプタ〉, r
```

loadのRオプションとrunのRオプションは，機能が全く同一で，指定したファイルのプログラムをロードしたのち，そのプログラムの実行をします．ここまでは

run 〈ファイルディスクリプタ〉

と同じですが，Rオプションが付くと，すべてのオープンされているデータファイルは，クローズせずに，オープンされたままになります．

この2つのRオプション付き命令は，プログラムのチェイン（「11.3.9　プログラムのチェイン」を参照）にも使えます．

bload文は，機械語プログラムでのRオプションなので，データファイルのクローズ，オープンは関係ありません．ただロードした後，プログラムを実行するというだけのRオプションです．

11.3.9　プログラムのチェイン

プログラムが大きくて，全部がメモリに入りきらない場合は，次のような方法が使われます．

(1)　プログラムを幾つかのモジュールに分けます．

(2)　それぞれ別のファイル名で，ディスクにセーブしておきます．

(3)　必要なモジュールのみをメモリにロードして実行します．

(4)　その後，自動的に次のモジュールをロードして実行を続けます．

このような方法のことを“プログラムをチェインにする”と呼びます．

この方法は，11.3.8で述べたload, runのRオプションを使っても可能ですが，この方法にもっと適した命令が用意されています．それは，chain命令です．

```
chain 〔merge〕〈ファイルディスクリプタ〉
          〔,〔行番号〕〔, all〕〔, delete 〈範囲〉〕〕
```

chain命令は，次のような命令です．

(1)　ファイルディスクリプタで指定したプログラムをロードして実行します．

(2)　ロードする前にオープンされているファイルはクローズされません．

(3)　〔行番号〕は，ロードしたプログラムの実行開始行を指定します．省略した場合はプログラムの最初から実行します．

(4)　ロードする前の変数をそのまま使用することもできます．その場合は，〔all〕オプションを指定してすべての変数が使えるようにするか，common文を使って，必要な変数だけを使えるようにするかのどちらか一方の指定が必要です．（詳細は「BASICリファレンスマニュアル」のcommon文を参照して下さい．）

(5)　〔merge〕オプションは，ロードする前にメモリ上にあるプログラムと指定したプログラム（アスキー形式）をマージして，指定した行番号から実行させせます．

(6)　〔delete〕オプションは，mergeオプション指定の時のみ付けられ，マージする前にメモリ上のプログラムの指定した範囲を削除します．

例えば，次のように指定します．

（例）　chain "work"

chain "work", 100, all

chain merge "program", 1000, all, delete 10000-20000

詳細は，「BASICリファレンスマニュアル」を参照してください．

11.3.10　ディスク入出力用の関数

ディスク入出力を効率よく行うために次に示す関数が用意されています．詳細は「BASICリファレンスマニュアル」を参照してください．

eof	シーケンシャルファイルの終りを検出
mki$	整数を2バイトの文字列へ変換
mks$	単精度実数を4バイトの文字列へ変換
mkd$	倍精度実数を8バイトの文字列へ変換
cvi	2バイトの文字列を整数型の数値に変換
cvs	4バイトの文字列を単精度実数型の数値に変換
cvd	8バイトの文字列を倍精度実数型の数値に変換
lof	ディスクファイルの大きさをセクタ数で表す
dskf	ディスクに関する情報を返す
input$	指定した長さの文字列をファイルから読み込む
loc	ランダムファイルの場合は最後に読み書きしたレコードのレコード番号，シーケンシャルファイルの場合は今まで読み書きしたレコード数を返します．
attr$	ディスクまたはファイルの属性を返す
fpos	最終レコードの物理セクタ番号を返す

11.3.11　DSKI$とDSKO$

ディスクに対して直接的な入出力を行う場合，dski$関数およびdsko$ステートメントを使用します．非常に危険なステートメントです．したがって，このステートメントを使用するには，十分注意をして下さい．このステートメントを実行する前に，使用するディスクのバックアップを作っておいてください．

(1)　**dski$**

ディスクの内容を読み込む場合，これまで説明してきたステートメントでは，ファイル名かファイル番号でその内容を指定しましたが，dski$は物理アドレス（ドライブ番号，サーフェス番号，トラック番号，セクタ番号）で指定した内容をランダムアクセス用バッファに読み込む関数です．

dski$（〈ドライブ番号〉，〈サーフェス番号〉，〈トラック番号〉，〈セクタ番号〉）

dski $は１セクタのデータ（256バイト）を文字列として返します．ところが文字変数の最大の長さは255文字なので，256バイト全部を読み込む為に，フィールド変数を使用します．

（例）
```
10 field #0, 128 as a$, 128 as b$
20 d$ = dski$ (2, 1, 39, 16)
```

この例では，ドライブ番号２，サーフェス１，トラック39，セクタ16の１セクタの前半の128バイトをa$に，後半の128バイトをb$に読み込みます．またそのセクタの前から255バイトはd$に代入されます．

このようにdski$は，ランダムファイルと似たような使い方をします．

(2)　dsko$

dsko$はdski$と同様に，＃０のランダムアクセス用バッファを使用します．field#0によって指定したランダムアクセス用バッファの内容を指定したセクタに書き込みます．(field#0の指定は「11.4.3」の中のfield文の説明を参照して下さい.)

dsko$　〈ドライブ番号〉,〈サーフェス番号〉,〈トラック番号〉,〈セクタ番号〉

（例）　dsko$　２，１，39，16

この例では，ランダムファイルの場合と同じように，＃０のランダムアクセス用バッファの内容を，ドライブ２，サーフェス１，トラック39，セクタ16の１セクタに書き込みます．

11.4　データの入出力

11.4.1　シーケンシャルファイルとランダムファイル

データファイルにはシーケンシャルファイルとランダムファイルの２種類があります．

(1)　シーケンシャルファイル

このファイルの特徴としては，次の事があげられます．

①　データの読み書きを行う場合は，必ずファイルの先頭から順次行います．
②　ファイル内の一部のデータだけを，書き変えることはできません．
③　ファイルの変更は，ファイルの最後に新たなデータを追加することだけです．
④　比較的簡単に使えます．
⑤　１つのレコード（データ処理の単位で，レコードの集りをファイルと呼びます）の大きさは，自由に変えられます．

(2)　ランダムファイル

このファイルの特徴としては次の事があげられます．

①　ファイル内の各レコードにランダムにアクセスできるので，データ（レコード）の読み書きを行う場合はファイルの先頭から順次行わなくてもよい．
②　ファイル内の一部のデータだけを，書き変えることができます．
③　ファイルの最後に，新たなデータを追加することもできます．
④　シーケンシャルファイルに比べて，プログラムがやや複雑になります．
⑤　１つのレコードの大きさはすべて同じです．

また，シーケンシャルファイルに対する入出力をシーケンシャルファイルアクセス，ランダムファイルに対する入出力をランダムファイルアクセスと呼びます．

11.4.2 シーケンシャルファイルに対するデータの入出力

シーケンシャルファイルアクセスの手順は次のとおりです.

① ファイルをOPENする.

② データを読み書きする.

③ ファイルをCLOSEする

OPENモード

ファイルをOPENする際，ファイルに対する処理形態（OPENモード）を宣言します.

(1) OUTPUTモード

新規にファイルを作成し，先頭から順次データを書き込んでいきます.

(2) INPUTモード

既存のファイルの先頭から順次データを読み込んでいきます.

(3) APPENDモード

既存のファイルの最後にデータを追加していきます.

OPENモードと入出力の関係

(1) OUTPUTモード

一般にデータの出力だけが行えます.

ただし，ディスクファイルに限り入力も可能です.

OUTPUTモードでOPENされたファイルに対して入力要求が行われた場合，その時の入出力バッファの内容が入力データとなります.

(2) INPUTモード

データの入力だけが行えます.

(3) APPENDモード

データの出力だけが行えます.

詳細については「BASICリファレンスマニュアル」を参照してください.

文	OPEN，CLOSE，PRINT#，WRITE#，PRINT#USING，INPUT#，LINE INPUT#
関　数	EOF，LOF，LOC，INPUT$

シーケンシャルファイルアクセスの概要

ディスクファイルを使用したシーケンシャルファイルアクセスの例を示します.

(1) シーケンシャルファイルの作成

キーボードから名前や年齢等の情報を得，それをシーケンシャルファイルに記録するプログラムです.

```
10 OPEN "TEST" FOR OUTPUT AS #1
20 INPUT "名前"; N$
30 IF N$="" THEN END
40 INPUT "出身地"; B$
50 INPUT "年齢"; A
```

```
60 INPUT "性別"; S$
70 INPUT "登録番号" ; N
80 PRINT #1, N$ ; ", " ; B$ ; ", " ; A ; S$ ; ", " ; N
90 GOTO 20
```

OUTPUTモードでOPENします.（ファイルは新規作成となります）
シーケンシャルファイルにデータを書き出す場合，各データは適切な区別り記号で区切らなければなりません（行番号80のPRINT #文を見てください）
数値型データの区切り：空白，コンマ，キャリッジリターン（CHR $(13)）
文字型データの区切り：コンマ，キャリッジリターン（CHR $(13)）
このプログラムを実行してみます.

```
run
  名前？ K.YAMADA
  出身地？ 東京都
  年齢？ 32
  性別？ 男
  登録番号？ 1132
  名前？ S.TANAKA
  出身地？ 大阪府
  年齢？ 26
  性別？ 女
  登録番号？ 6553
  名前？
  OK
```

この場合，ファイルには次の形式でデータが出力されます.

K. YAMADA，東京都，△32△男，△1132△ (CR)(LF) S. TANAKA,

大阪府，△26△女，△6553△ (CR)(LF)　　△は空白を意味します.

説明
① 日本語1文字は2バイト分の領域を占めます．たとえば“東京都”の場合6バイトを必要とします．また，前後にKIコードおよびKOコード（各2バイト）が挿入されます.
② この例の場合，文字型のデータはすべてコンマで区切られています（コンマはデータとして出力しなければなりません）.
③ 数値型のデータ（たとえば，△32△や△1132△）は空白で区切られています．数値型データの前には，符号が正の場合，自動的に1個の空白が生成されます．また，数値型データの終端にも必らず1個の空白が生成されます．したがって，数値型の前後に空白をデータとして出力してやる必要はありません.

④ 80 PRINT ＃１，N$；"，"；B$；"，"；A；S$；"，"；Nのように PRINT ＃文の終端にPRINT 制御記号（";"や","）が指定されていない場合，最終データ（この場合，Nの値）のうしろにキャリッジリターンとラインフィードが出力されます．ここまでがシーケンシャルファイルの１レコードです．参考までに，80 PRINT ＃１，N$；"，"；B$；"，"；A;S$；"，"；N；のように，PRINT ＃文の終端に"；"を指定した場合の出力イメージは次のようになります．

```
K. YAMADA，東京都，△32△男，△1132△ S. TANAKA，大阪府，
```

```
△26△女，△6553△
```

この場合でも各データは区切り記号で適切に区切られていますのでINPUT ＃文でこのデータを読む限り正しく読むことができます．
ただし，LINE INPUT ＃文で読む場合，うまく読めません．
LINE INPUT ＃文は(CR)(LF)までを１つのレコードとしてまとめて読みます．この場合(CR)(LF)が出力されていませんので正しく読むことはできません．この例の場合のPRINT ＃文はWRITE ＃文を使い次のように書き直すことができます．

```
80 WRITE ＃１，N$，B$，A，S$，N
```

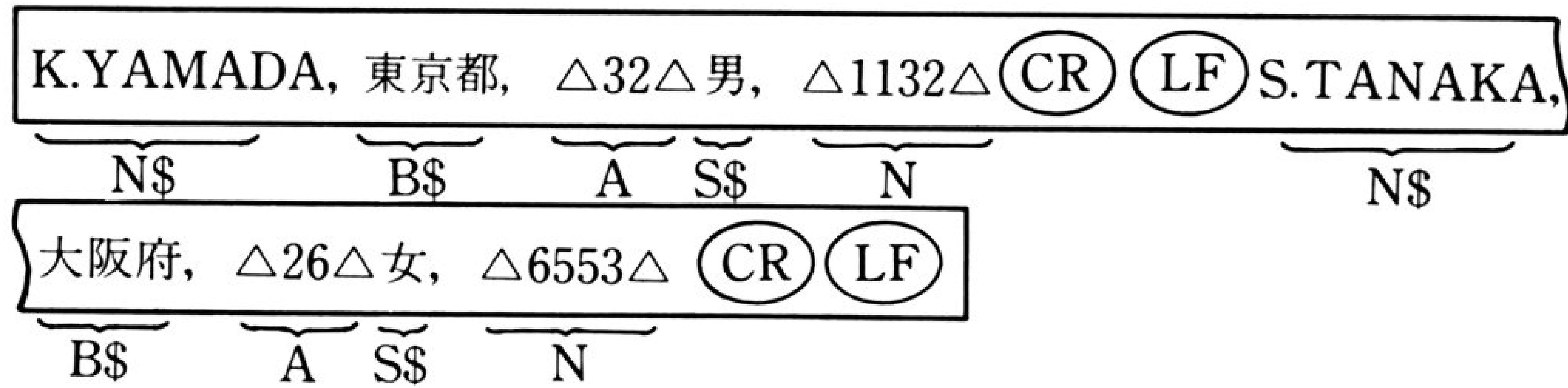
WRITE＃文は各データを自動的にカンマで区切って出力しますので，PRINT＃文のように "," をデータとして出力する必要はありません．

(2) シーケンシャルファイルからのデータの読み込み

前述のプログラムで作成したデータファイルを読んでみます．

```
10 OPEN "TEST" FOR INPUT AS＃１
20 IF EOF(＃１) THEN END
30 INPUT＃１，N$，B$，A，S$，N
40 PRINT N$；"  出身地："；B$；"  年齢：";A；" 性別："S$；"  登録番号："；N
50 GOTO20
run
K.YAMADA  出身地：東京都  年齢：32  性別：男  登録番号：1132
S.TANAKA  出身地：大阪府  年齢：26  性別：女  登録番号：6553
OK
```

ファイルはINPUT モードでOPEN します
EOF（＃１）はファイルの終端に達したことを通知してくれる関数です．
INPUT ＃文でデータを読む場合，入力されるデータの型と受ける変数の型が一致しなければなりません．



文字型のデータはコンマとキャリッジリターンコードが区切り記号となります．数値型のデータは空白，コンマ，キャリッジリターンコードが区切り記号となります．

(3) シーケンシャルファイルに対するデータの追加

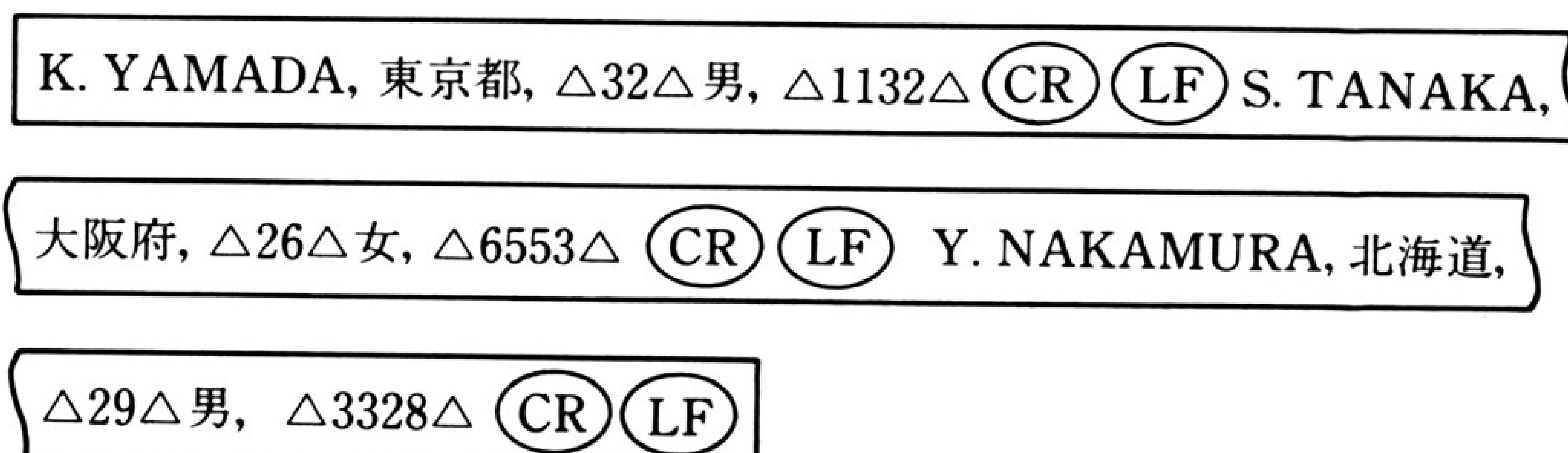
APPENDモードでOPENします．

```
10  OPEN "TEST" FOR APPEND AS#1
20  INPUT "名前"; N$
30  IF N$=""THEN END
40  INPUT "出身地"; B$
50  INPUT "年齢"; A
55  INPUT "性別"; S$
60  INPUT "登録番号"; N
70  PRINT#1, N$;", ";B$;", "A ; S$;", ";N
80  GOTO 20
```

レコードを1個追加します．

```
run
名前？  Y.NAKAMURA
出身地？北海道
年齢？  29
性別？  男
登録番号？  3328
名前？
OK
```

ファイルの内容は次のようになります．



このデータを読み直します.

```
10  OPEN "TEST" FOR INPUT AS#1
20  IF EOF (#1) THEN END
30  INPUT#1, N$, B$, A, S$, N
40  PRINT N$;"  出身地:";B$;"  年齢:";A;"性別:";S$;"  登録番号:";N
50  GOTO 20
run
K.YAMADA出身地:東京都  年齢:32  性別:男  登録番号:1132
S.TANAKA出身地:大阪府  年齢:26  性別:女  登録番号:6553
Y.NAKAMURA出身地:北海道  年齢:29  性別:男  登録番号:3328
OK
```

11.4.3 ランダムファイルに対するデータの入出力

ランダムファイルアクセスはディスクファイルだけに許されるファイル操作です.
ランダムファイルアクセスの手順は次のとおりです.

① ファイルをOPENする.
② 入出力バッファにデータをセットする，あるいはファイルから任意のレコードを入出力バッファに読み込む.
③ 入出力バッファの内容をファイルの任意のレコードに書き出す，あるいは入出力バッファに読み込んだレコードの内容を取り出す.
④ ファイルをCLOSEする.

OPENモード

ランダムファイルアクセスを行う場合，OPEN文にはOPENモードを指定しません.

ランダムファイルはレコードの入出力が自由に行えるファイルですから，シーケンシャルファイルの場合のように入出力の形態を宣言する必要はありません.

ランダムファイルアクセスで使用する命令

文	OPEN, CLOSE, GET#, PUT#, FIELD, LSET RSET
関数	LOC, LOF, MKI$, MKS$, MKD$, CVI CVS, CVD

ランダムファイルのレコードサイズとランダムアクセス用バッファ

(1) ランダムファイルのレコードサイズ

ランダムファイルを構成する各レコードのレコードサイズはすべて同じサイズです.

ランダムファイルのレコードサイズは256バイト固定です.

(2) ランダムアクセス用バッファ

ランダムファイルに対するレコードの入出力はランダムアクセス用バッファを介して行われます.

ランダムファイルアクセスの概要

ランダムファイルの作成/更新の方法を具体的に説明します.

(1)　ランダムファイルの作成

```
①  10 FIELD＃1，30 AS NN$, 16 AS BB$, 2 AS AA$, 6 AS SS$, 4 AS NO$
②  20 OPEN "DATA" AS＃1
    25 *NEXTP
    30 INPUT "名前" ；N$
    40 IF N$="" THEN END
    50 INPUT "出身地"；B$
    60 INPUT "年齢"；A%
    70 INPUT "性別"；S$
    80 INPUT "登録番号"；N
③  90 LSET NN$=N$：LSET BB$=B$：LSET AA$=MKI$（A%）
④  100 LSET SS$=S$：LSET NO$=MKS$（N）
⑤  110 PUT＃1，N
    120 GOTO*NEXTP
```

①　ランダムファイルに対するレコードの入出力はランダムアクセス用バッファを介して行います．行番号10のFIELD文はランダムアクセス用バッファに対するフィールド定義を行うための命令です．ランダムファイルの1レコードは256バイトですから，256バイトの範囲でフィールド定義が可能です．

このプログラムの場合，次のようなフィールド定義となります．

NN$	BB$	AA$	SS$	NO$	未使用
30バイト	16バイト	2バイト	6バイト	4バイト	198バイト

各フィールドには文字型の変数名を与えます．各フィールドに対するデータの設定あるいは各フィールド内容の参照はこの変数名（フィールド変数と呼びます）で行います．

②　ランダムファイルに対するOPENはOPENモードを指定しません．

③，④　各フィールドに対してデータを設定します．

フィールドに対するデータの設定はこのように必ずLSETあるいはRSET文を使用します．通常の代入文ではフィールドに対するデータ設定は行えません．

LSETはフィールドに対し左づめでデータをセットします．

RSETは右づめです．いずれも余りに空白がつめられます．

N$，B$，S$は文字型のデータですから，そのままフィールド変数に代入（フィールドに設定）できます．

A%，Nは数値型のデータですから，そのままフィールド変数に代入することはできません．

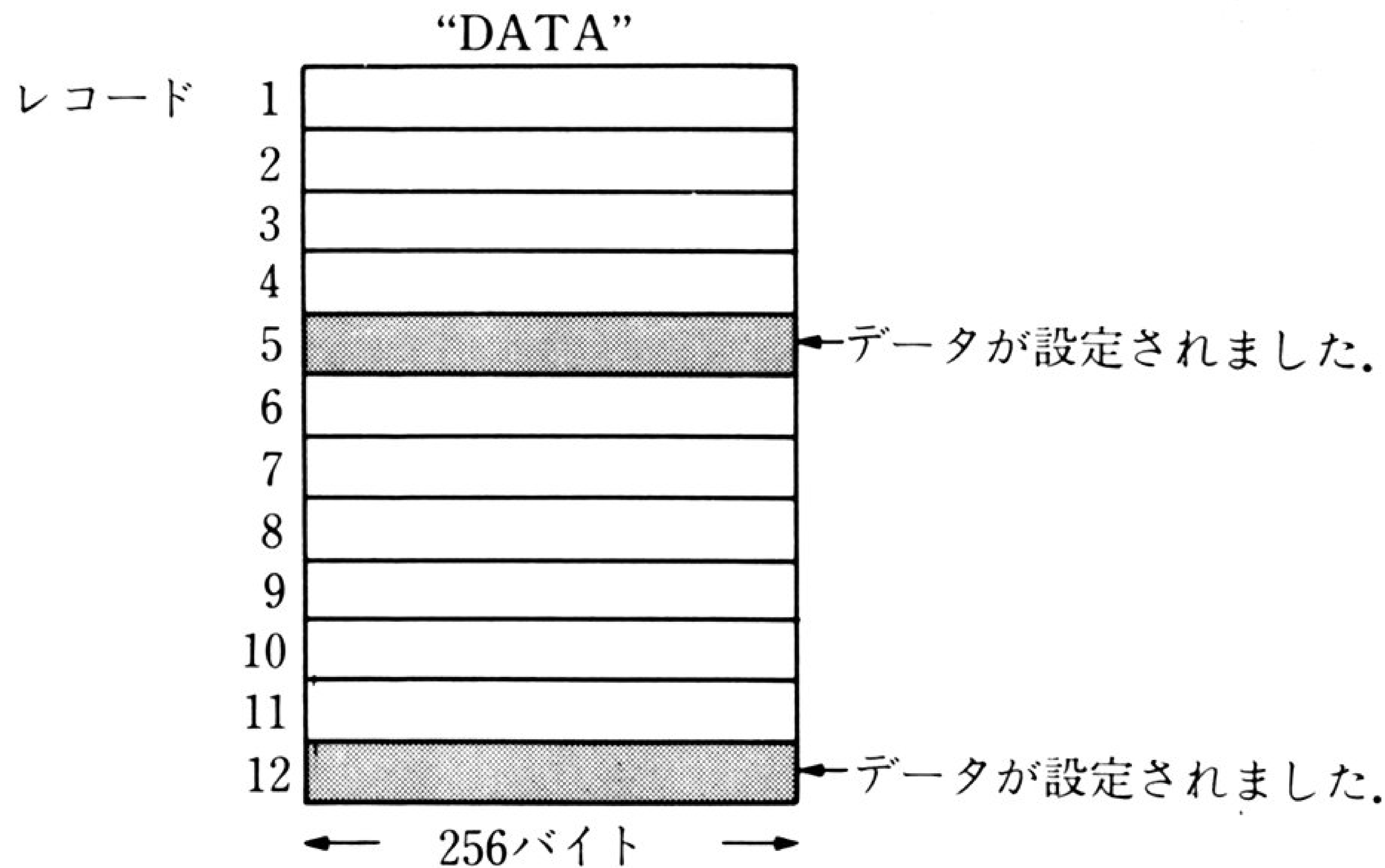
```
LSET AA$=MKI$（A%）
LSET NO$=MKS$（N）
```

のように特殊な関数を使用し，一時的に変数の型を文字型にします．

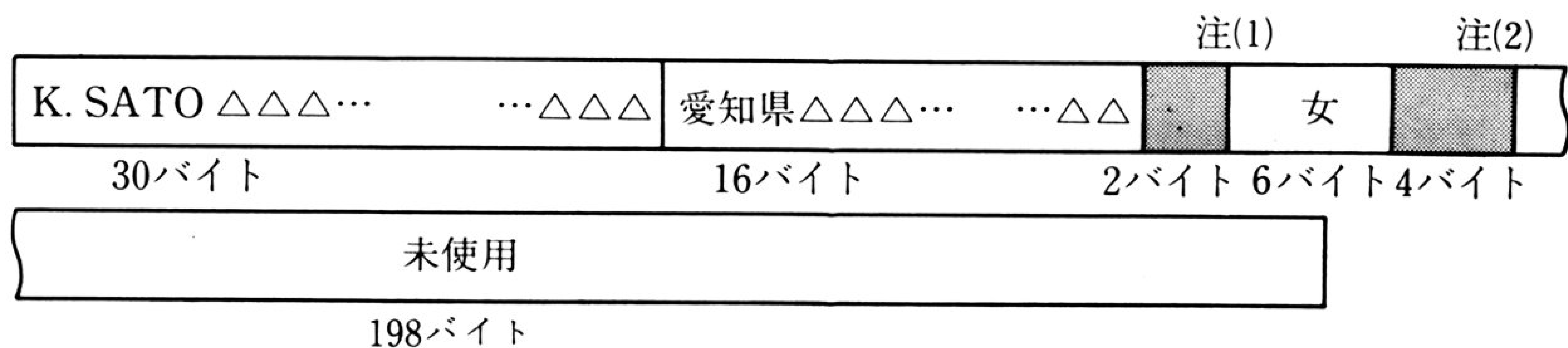
MKI＄は整数型のデータに2バイトの文字型データの属性を与えます．
MKS＄は単精度実数型のデータに4バイトの文字型データの属性を与えます（倍精度実数型のデータに対してはMKD＄関数を使用します.）
これらの関数は数値型のデータを一時的に文字型の属性で扱えるようにするだけで，数値データそのものを文字に変換するわけではありません．したがって，LSET AA＄＝MKI＄（A％）はA％の値をそのまま（数値の内部形式のまま）フィールドに設定します．
整数型のデータは内部では2バイトで表現されていますので，フィールドの長さも2バイトでよいのです．
同様に単精度実数型データは4バイト，倍精度実数型は8バイトです．
（シーケンシャルファイルにおいては，数値データは文字に変換されてファイルに出力されます．たとえば，A％＝32760：PRINT＃1，A％において，A％の値は内部では2バイトのデータですが，ファイルに出力される場合，“△32760△”という7文字の文字で出力されます．△は空白を意味します）．
⑤　ランダムアクセス用バッファの内容をN番目のレコードとしてファイルに出力します．
次のようにプログラムを実行してみます．

```
run
名前？　K.MATSUMOTO
出身地？　東京都
年齢？　28
性別？　男
登録番号？　12
名前？　K.SATO
出身地？　愛知県
年齢？　22
性別？　女
登録番号？　5
名前？
OK
```

ファイル“DATA”には12番目と5番目のレコードが作成されました．



たとえば，5番目のレコードの内容は次のようになっています.



注(1)　22の整数内部形式が格納されている.

注(2)　5の単精度実数内部形成が格納されている.

(△は空白を意味します．また，日本語の1文字は2バイトを占め，前後にKIコードおよびKOコード（各々2バイト）が挿入されます.)

(2)　ランダムファイルの更新

前述のプログラムで作成したファイル"DATA"の5番目のレコードの年齢フィールドのみを更新します.

```
①  10  FIELD#1, 30 AS NN$, 16 AS BB$, 2 AS AA$, 6 AS SS$, 4 AS NO$
②  20  OPEN "DATA" AS #1
    30  *NEXTP
    40  INPUT "更新対象者の番号", N
    50  IF N=0  THEN END
③  60  GET#1, N
    70  PRINT "更新対象者の現状は次のとおりです"
    80  PRINT "名前:"; NN$
    90  PRINT "出身地:"; BB$
④  100  PRINT "年齢:"; CVI(AA$)
    110  PRINT "性別:"; SS$
```

```
    120 PRINT "登録番号：";CVS(NO$)
    130 INPUT "年齢を入力してください";A%
⑤  140 LSET AA$=MKI$(A%)
⑥  150 PUT #1, N
    160 PRINT
    170 GOTO *NEXTP
run
更新対象者の番号  5
更新対象者の現状は次のとおりです
名前：K.SATO
出身地：愛知県
年齢：22
性別：女
登録番号：5
年齢を入力してください？  24
更新対象者の番号  0
OK
```

① データを作成した時と同様のフィールド定義を行います．

② OPENモードは指定しません．

③ 対象レコード（レコード番号は5）の内容をランダムに読み込みます．

④ 読み込んだデータを参照する場合，文字型データ（NN$, BB$, SS$）はそのままフィールド変数名で引用することができます．
数値データは内部形式のまま格納されていますのでフィールド変数名（文字型変数）で直接引用することはできません．CVI（AA$），CVS（NO$）のように一時的に数値データの型に変換して引用しなければなりません．（倍精度実数として格納されている場合，CVD関数を使用します）

⑤ フィールドAA$のみデータを更新します．

⑥ レコードを書きもどします．

第12章
ディスクの構造

この章ではディスクのファイル構造について説明します．

12.1 ディスクの物理構造

使用できるディスクの物理構造を説明します．

ディスクの型	サーフェス番号	トラック数	トラック番号	セクタ数	セクタ番号	ディスク形式参照番号
3.5インチおよび5インチ両面倍密度倍トラック	0～1	80	0～79	16	1～16	図12.1.1
3.5インチおよび5インチ高密度 8インチ両面倍密度	0～1	77	0～76	26	1～26	図12.1.2
固定ディスク（5MB）	0～3	153	0～152	33	1～33	図12.1.3
固定ディスク（10MB）	0～3	310	0～309	33	1～33	図12.1.4
固定ディスク（20MB）	0～7	308	0～307	33	1～33	図12.1.5

表の見方
1．トラック数は面あたりの数値，セクタ数はトラックあたりの数値です．
2．読み書き単位は，セクタで行います．セクタの大きさは，256バイト共通です．
3．3.5インチ，5インチ高密度および8インチ両面倍密度ディスクの場合，サーフェス番号0，トラック番号0は1セクタ128バイトでフォーマットされています．
4．固定ディスクの場合，最内周のトラックは診断用に予約されています．
5．固定ディスク（20MB）の場合，307トラックのサーフェス4～7は使用できません．

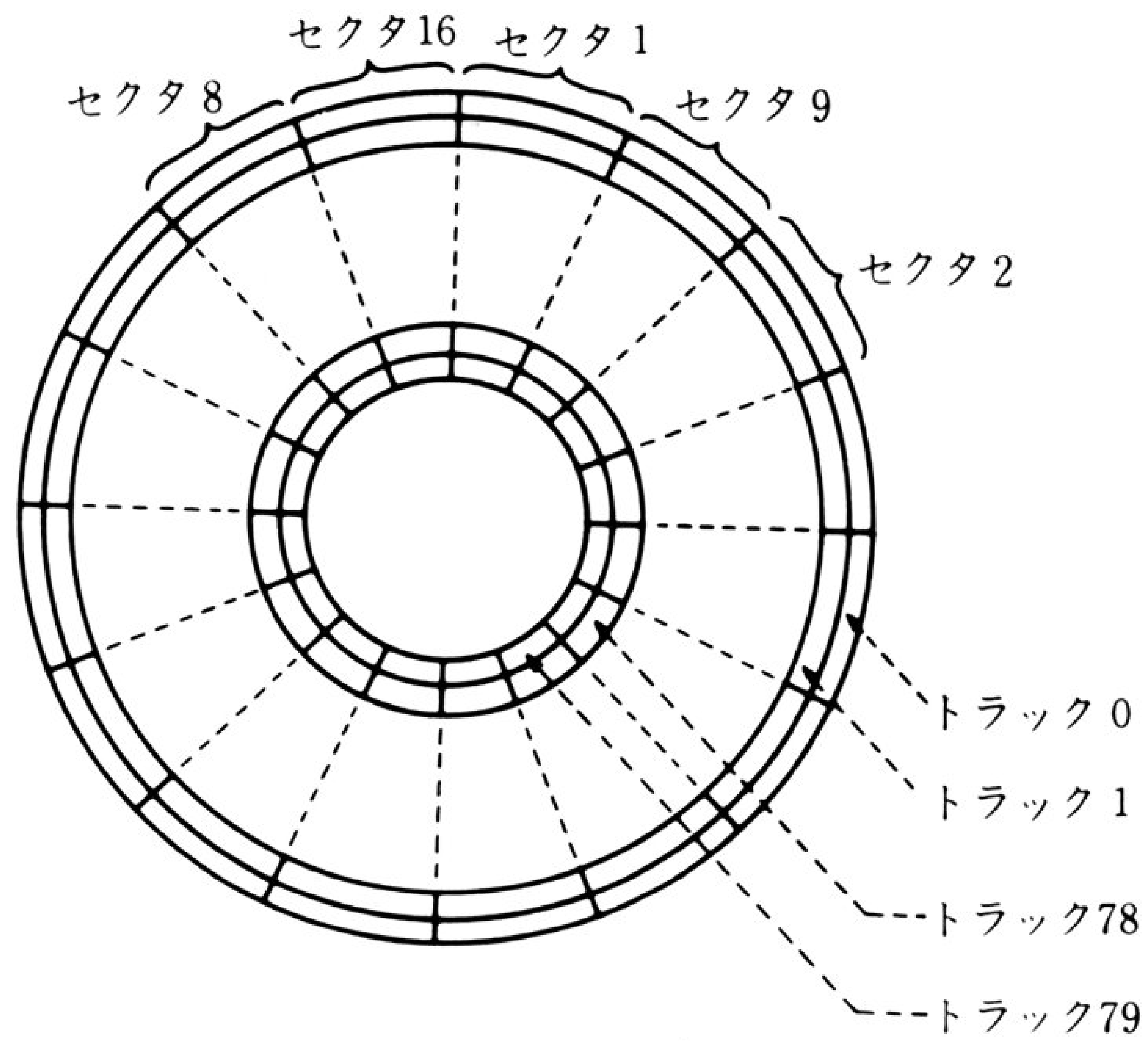


図12.1.1　3.5インチおよび5インチ両面倍密度倍トラック
フロッピィディスク

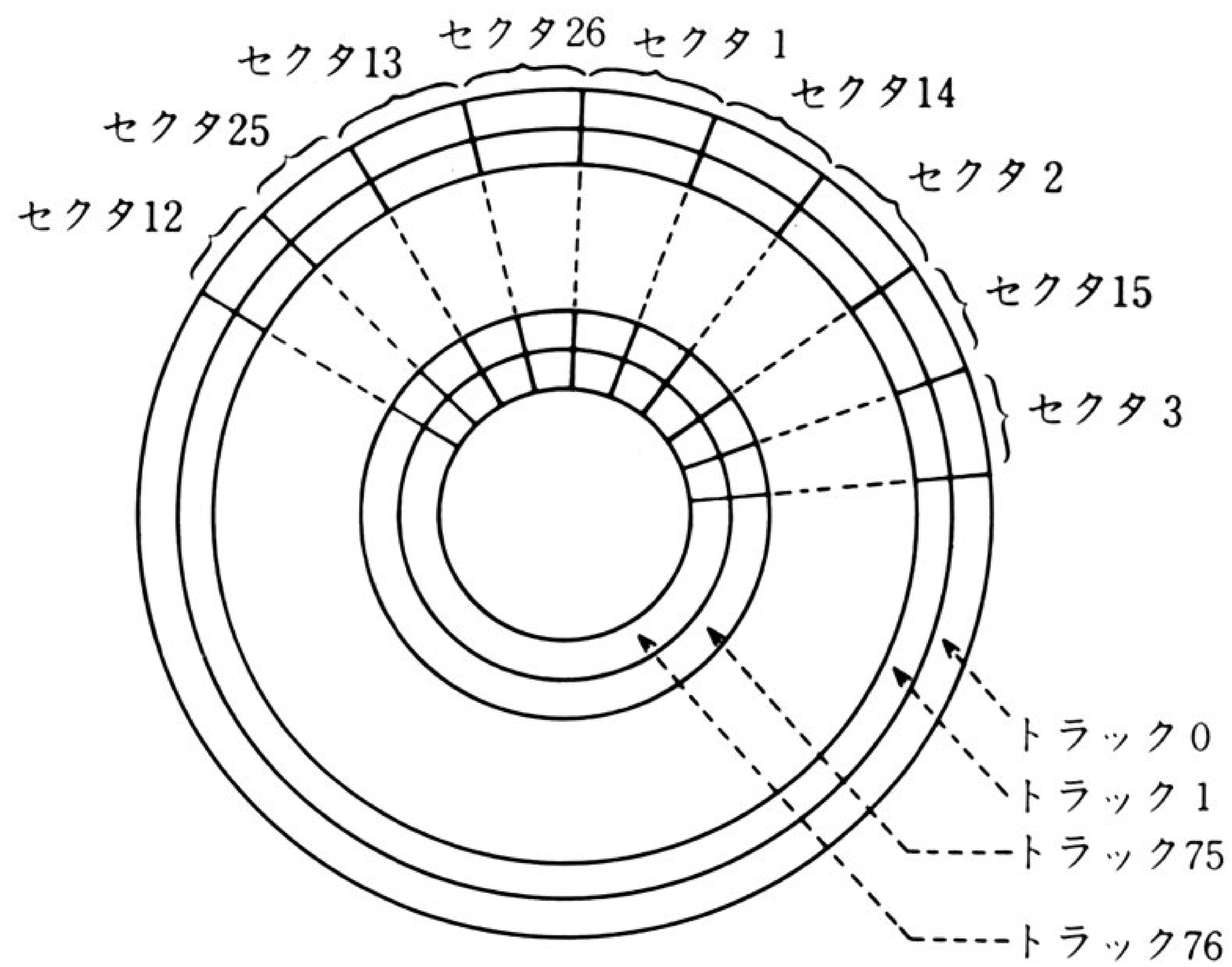


図12.1.2　3.5インチ，5インチ高密度フロッピィディスク
および8インチ両面倍密度フロッピィディスク

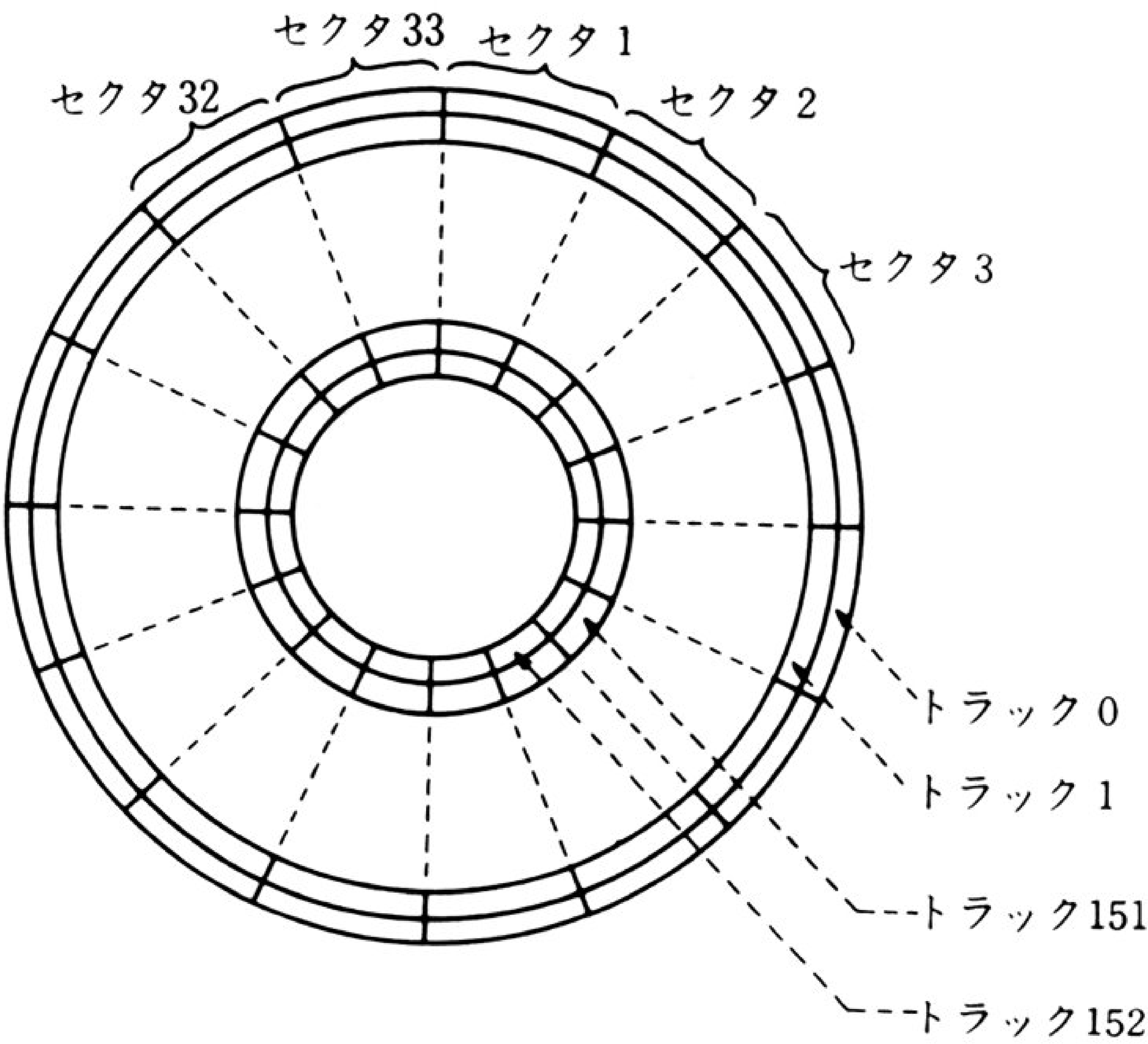


図12.1.3　固定ディスク（5MB）

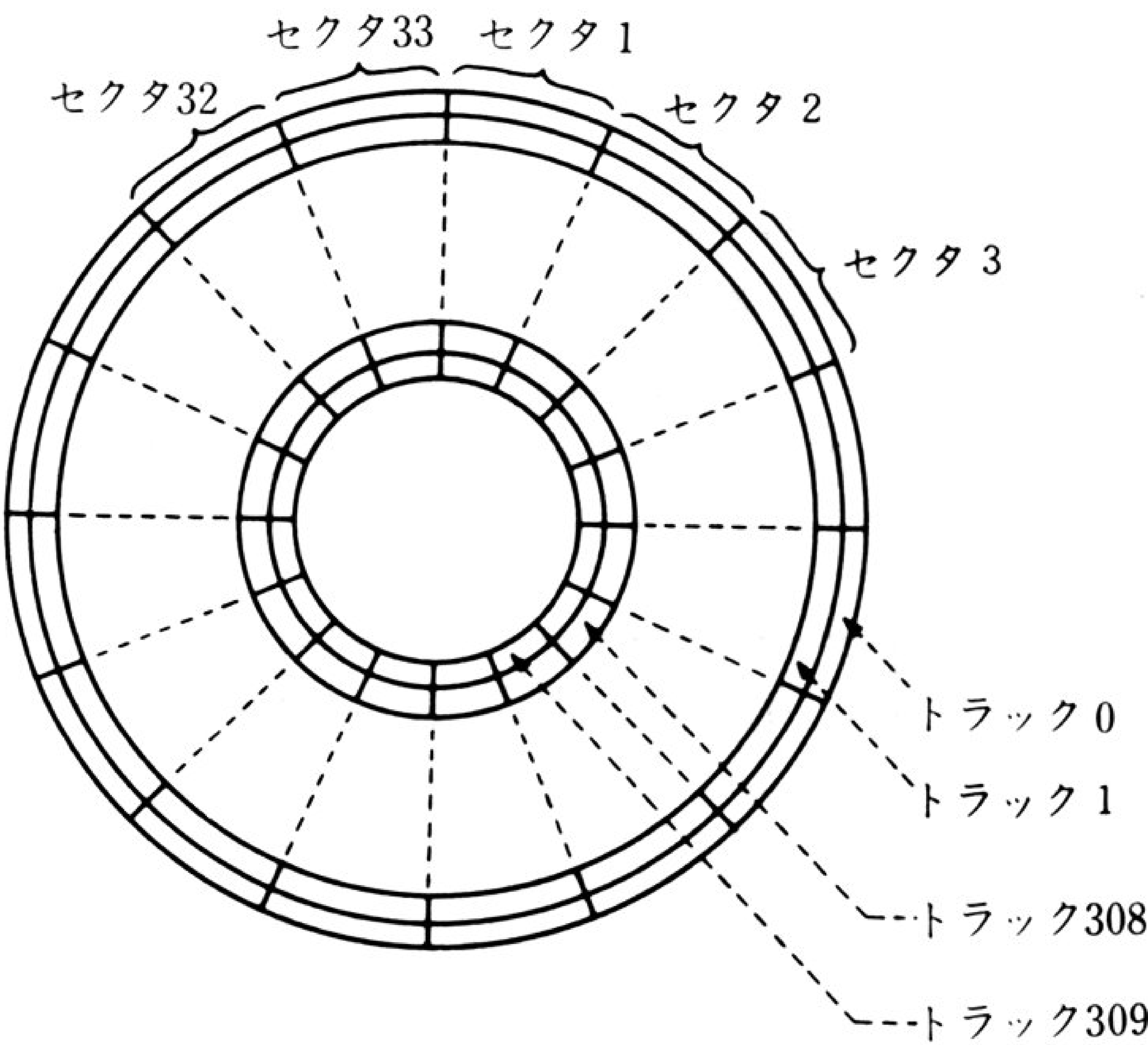


図12.1.4　固定ディスク（10MB）

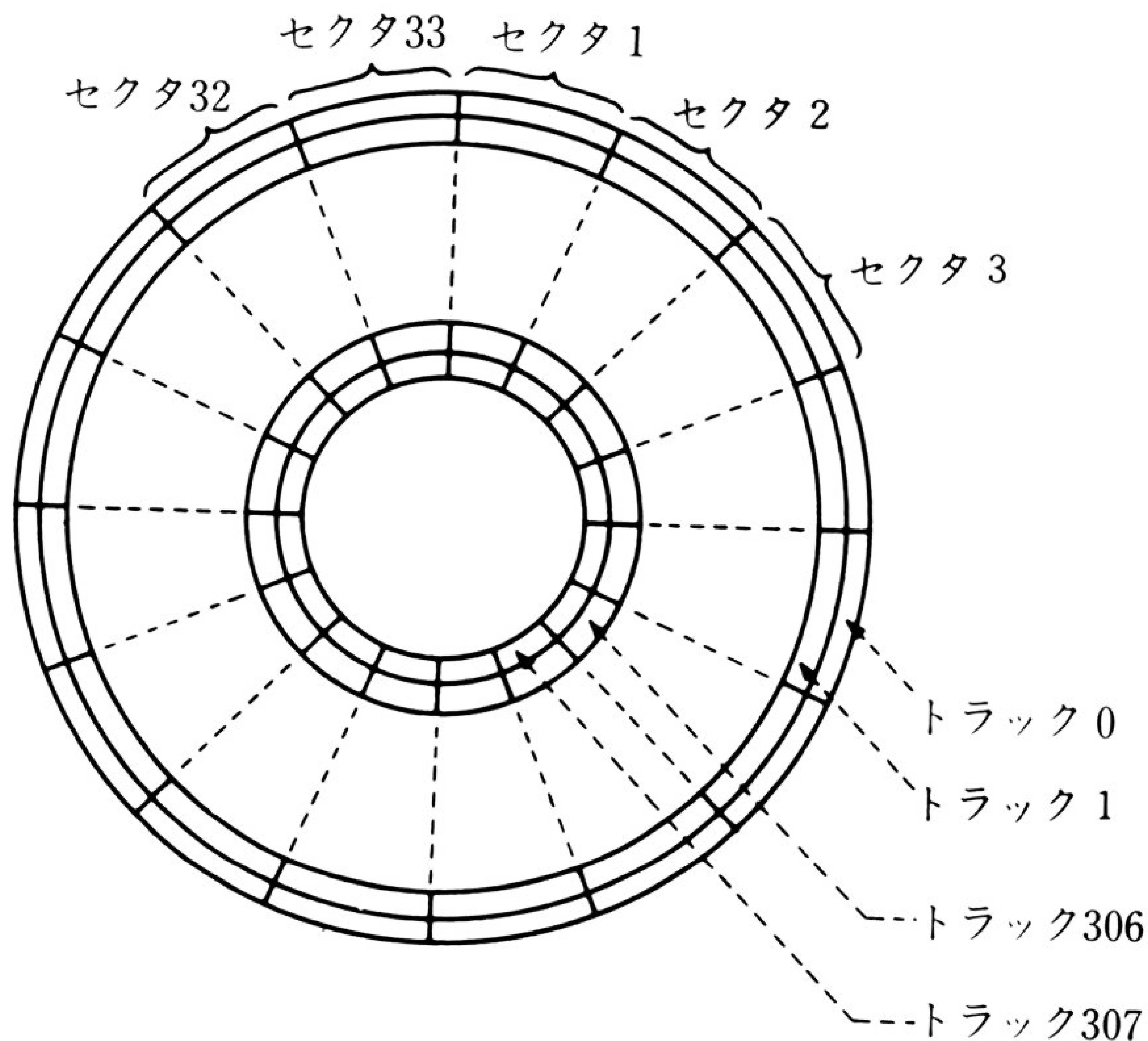
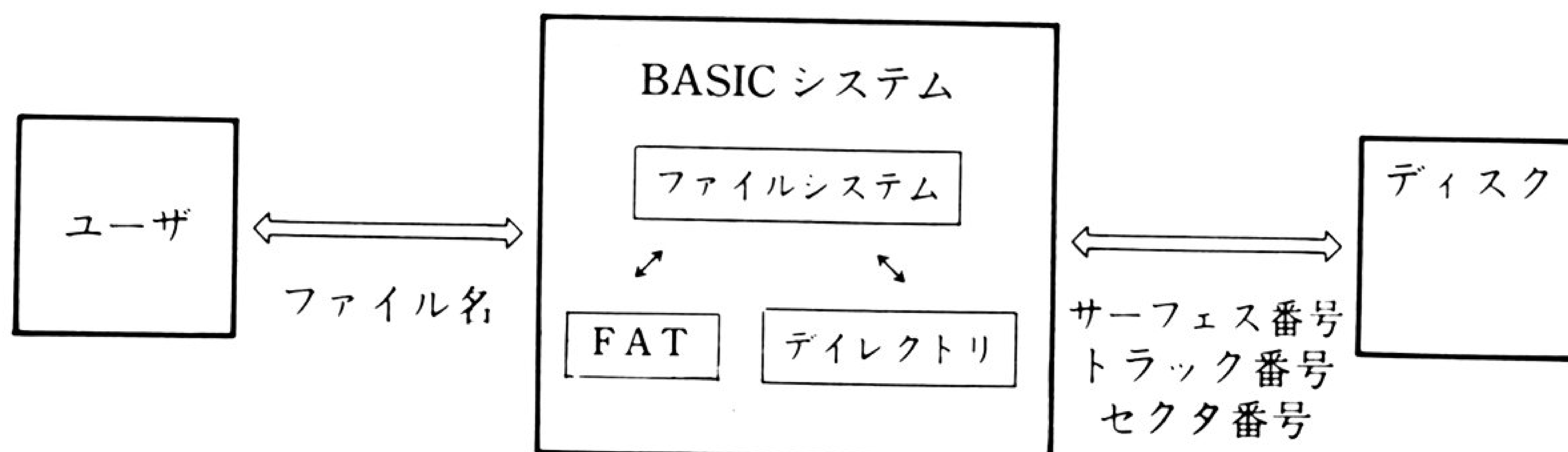


図12.1.5　固定ディスク（20MB）

12.2 クラスタ，FAT，ディレクトリ

通常，BASICでディスクファイルを扱う場合，サーフェス番号，トラック番号，セクタ番号等の物理的なディスク管理単位を意識する必要はありません．ファイル名とそのファイルの内容をディスク上に記憶したり，ディスクからメモリへ読み出したりする場合に，実際のディスク上の物理アドレス（サーフェス番号，トラック番号，セクタ番号）に必要な変換をするのはシステムの仕事です．



ファイル名とファイルのすべての情報を管理するために，2つの管理表があります．“ディレクトリ”と“FAT”です．

ディレクトリはそのディスクに格納されているすべてのファイルの名前，属性，格納されている

場所（そのファイルが格納されている先頭のクラスタ番号）を管理しています．

FATは，ディスクの領域が使用されているか，使用されていないかの状態を示すと同時に，それぞれのファイルがどの領域を使用しているかを示す管理表です．

ディスクの読み書きの最小単位は1セクタですが，領域の確保・解放，または使用・未使用の最小単位は**“クラスタ”**という，ディスク媒体によって決まる大きさです．それぞれの媒体のクラスタの大きさと総数は次のようになっています．

ディスクタイプ	クラスタの大きさ	クラスタ総数
3.5インチ， 5インチ両面倍密度倍トラックフロッピィディスク	16セクタ	160
3.5インチ，5インチ高密度フロッピィディスク 8インチ両面フロッピィディスク	26セクタ	154
固定ディスク（5MB）	33セクタ	612
固定ディスク（10MB）	33セクタ	1240
固定ディスク（20MB）	33セクタ	2460

クラスタ番号と物理アドレスとの対応

ファイルシステムを使わずに直接ディスクから読み書きするためには，ディレクトリ，FATを参照して，BASICとかかわりのない領域を定める必要があります．ディレクトリ，FATはクラスタ番号によってディスク上のアドレスを表わしています．一方，直接ディスクを扱う入出力関数dski$，dsko$は物理アドレス（サーフェス番号，トラック番号，セクタ番号）を使います．それゆえ，クラスタ番号を物理アドレスに変換する必要があります．このような場合に，クラスタ番号と物理アドレスの対応表が役に立ちます．以下に，これを示します．

(a)　3.5インチおよび5インチ両面倍密度倍トラックフロッピィディスクの場合

クラスタ番号	物理アドレス		
	サーフェス番号	トラック番号	セクタ番号
0	0	0	1～16
1	1	0	1～16
2	0	1	1～16
3	1	1	1～16
4	0	2	1～16
⋮	⋮	⋮	⋮
158	0	79	1～16
159	1	79	1～16

(b) 3.5インチ，5インチ高密度フロッピィディスクおよび8インチ両面倍密度フロッピィディスクの場合

クラスタ番号	物理アドレス		
	サーフェス番号	トラック番号	セクタ番号
0	0	0	1～26
1	1	0	1～26
2	0	1	1～26
3	1	1	1～26
4	0	2	1～26
⋮	⋮	⋮	⋮
152	0	76	1～26
153	1	76	1～26

(c) 固定ディスクの場合

5MB用

クラスタ番号	物理アドレス		
	サーフェス番号	トラック番号	セクタ番号
0	0	0	1～33
1	1	0	1～33
2	2	0	1～33
3	3	0	1～33
4	0	1	1～33
⋮	⋮	⋮	⋮
610	2	152	1～33
611	3	152	1～33

10MB用

クラスタ番号	物理アドレス		
	サーフェス番号	トラック番号	セクタ番号
0	0	0	1～33
1	1	0	1～33
2	2	0	1～33
3	3	0	1～33
4	0	1	1～33
⋮	⋮	⋮	⋮
1238	2	309	1～33
1239	3	309	1～33

20MB用

クラスタ番号	物理アドレス		
	サーフェス番号	トラック番号	セクタ番号
0	0	0	1～33
1	1	0	1～33
2	2	0	1～33
3	3	0	1～33
4	4	0	1～33
⋮	⋮	⋮	⋮
2454	6	306	1～33
2455	7	306	1～33
2456	0	307	1～33
2457	1	307	1～33
2458	2	307	1～33
2459	3	307	1～33

12.3 トラックの割り当て

(1) 3.5インチおよび5インチ両面倍密度倍トラックフロッピィディスク

(a) システムディスク

サーフェス番号	トラック番号	セクタ番号	内容
0	0	1～2	IPL
	0	3～16	DISK code
	1～27	すべて	DISK code
	28～39	すべて	ユーザ領域
	40	1～12	ディレクトリ
	40	13	ID
	40	14～16	FAT
	41～79	すべて	ユーザ領域
1	0～27	すべて	DISK code
	28～79	すべて	ユーザ領域

(b) ユーザディスク

サーフェス番号	トラック番号	セクタ番号	内　　　容
0	0 ～39	すべて	ユーザ領域
	40	すべて	システムディスクと同じ
	41～79	すべて	ユーザ領域
1	0 ～79	すべて	ユーザ領域

(2) 3.5インチ，5インチ高密度フロッピィディスクおよび8インチ両面倍密度フロッピィディスク

(a) システムディスク

サーフェス番号	トラック番号	セクタ番号	内　　　容
0	0（注1）	1 ～ 4	IPL
	0（注1）	5 ～26	システム予約
	1 ～17	すべて	DISK code
	18～34	すべて	ユーザ領域
	35	1 ～22	ディレクトリ
	35	23	ID
	35	24～26	FAT
	36～76	すべて	ユーザ領域
1	0	すべて	システム予約
	1 ～16	すべて	DISK code
	17～76	すべて	ユーザ領域

（注1） 1セクタ128バイトでフォーマットされている．
他のセクタはすべて256バイトでフォーマットされている．

(b) ユーザディスク

サーフェス番号	トラック番号	セクタ番号	内　　　内
0	0（注1）	1 ～26	システム予約
	1 ～34	すべて	ユーザ領域
	35	1 ～22	ディレクトリ
	35	23	ID
	35	24～26	FAT
	36～76	すべて	ユーザ領域
1	0	すべて	システム予約
	1 ～76	すべて	ユーザ領域

（注1） 1セクタ128バイトでフォーマットされている．
他のセクタはすべて256バイトでフォーマットされている．

(3)　固定ディスク（5MB）

(a)　システムディスク

サーフェス番号	トラック番号	セクタ番号	内　　容
0	0	すべて	IPL及びシステム予約
	1～7	すべて	DISK code
	8～75	すべて	ユーザ領域
	76	すべて	ディレクトリ
	77～151	すべて	ユーザ領域
	151	すべて	T & D
1～2	0～7	すべて	DISK code
	8～75	すべて	ユーザ領域
	76	すべて	ディレクトリ
	77～151	すべて	ユーザ領域
	152	すべて	T & D
3	0～7	すべて	DISK code
	8～75	すべて	ユーザ領域
	76	1	ID
	76	2～33	FAT
	77～151	すべて	ユーザ領域
	152	すべて	T & D

(b)　ユーザディスク

サーフェス番号	トラック番号	セクタ番号	内　　容
0	0	すべて	システム予約
	1～75	すべて	ユーザ領域
	76	すべて	ディレクトリ
	77～151	すべて	ユーザ領域
	152	すべて	T & D
1～2	0～75	すべて	ユーザ領域
	76	すべて	ディレクトリ
	77～151	すべて	ユーザ領域
	152	すべて	T & D
3	0～75	すべて	ユーザ領域
	76	1	ID
	76	2～33	FAT
	77～151	すべて	ユーザ領域
	152	すべて	T & D

注 意 上の表は媒体のすべてがBASICで使用され，かつ不良トラックがない場合を示しています．したがって，IPL, DISK codeおよびT&Dをのぞく部分はディスクフォーマット時のサイズ指定によりトラックの位置が異なってきます．

(4) 固定ディスク（10MB）

(a) システムディスク

サーフェス番号	トラック番号	セクタ番号	内　　容
0	0	すべて	IPL及びシステム予約
	1～7	すべて	DISK code
	8～154	すべて	ユーザ領域
	155	すべて	ディレクトリ
	156～308	すべて	ユーザ領域
	309	すべて	T & D
1～2	0～7	すべて	DISK code
	8～154	すべて	ユーザ領域
	155	すべて	ディレクトリ
	156～308	すべて	ユーザ領域
	309	すべて	T & D
3	0～7	すべて	DISK code
	8～154	すべて	ユーザ領域
	155	1	ID
	155	2～33	FAT
	156～308	すべて	ユーザ領域
	309	すべて	T & D

(b) ユーザディスク

サーフェス番号	トラック番号	セクタ番号	内　　容
0	0	すべて	システム予約
	1～154	すべて	ユーザ領域
	155	すべて	ディレクトリ
	156～308	すべて	ユーザ領域
	309	すべて	T & D
1～2	0～154	すべて	ユーザ領域
	155	すべて	ディレクトリ
	156～308	すべて	ユーザ領域
	309	すべて	T & D
3	0～154	すべて	ユーザ領域
	155	1	ID
	155	2～33	FAT

	156～308	すべて	ユーザ領域
	309	すべて	T & D

注 意 **上の表は媒体のすべてがBASICで使用され，かつ不良トラックがない場合を示しています．したがって，IPL, DISK codeおよびT&Dをのぞく部分はディスクフォーマット時のサイズ指定によりトラックの位置が異なってきます．**

(5) 固定ディスク（20MB）

(a) システムディスク

サーフェス番号	トラック番号	セクタ番号	内容
0	0	すべて	IPL及びシステム予約
	1～3	すべて	DISK code
	4～154	すべて	ユーザ領域
	155	すべて	ディレクトリ
	156～306	すべて	ユーザ領域
	307	すべて	T & D
1～3	0～3	すべて	DISK code
	4～154	すべて	ユーザ領域
	155	すべて	ディレクトリ
	156～306	すべて	ユーザ領域
	307	すべて	T & D
4～6	0～3	すべて	DISK code
	4～154	すべて	ユーザ領域
	155	すべて	ディレクトリ
	156～306	すべて	ユーザ領域
7	0～3	すべて	DISK code
	4～154	すべて	ユーザ領域
	155	1	ID
	155	2～33	FAT
	155～306	すべて	ユーザ領域

(b) ユーザディスク

サーフェス番号	トラック番号	セクタ番号	内容
0	0	すべて	システム予約
	1～154	すべて	ユーザ領域
	155	すべて	ディレクトリ
	156～306	すべて	ユーザ領域
	307	すべて	T & D

1～3	0～154 155 156～306 307	すべて すべて すべて すべて	ユーザ領域 ディレクトリ ユーザ領域 T & D
4～6	0～154 155 156～306	すべて すべて すべて	ユーザ領域 ディレクトリ ユーザ領域
7	0～154 155 155 156～306	すべて 1 2～33 すべて	ユーザ領域 ID FAT ユーザ領域

注　意　上の表は媒体のすべてがBASICで使用され，かつ不良トラックがない場合を示しています．したがって，IPL, DISK codeおよびT&Dをのぞく部分はディスクフォーマット時のサイズ指定によりディスクの位置が異なってきます．

12.4　ディレクトリ

(1)　ディレクトリの大きさとファイル数

ディレクトリは，ディスク上のファイルを管理するための表です．1つのファイルに対して，16バイトの制御情報を持っています．各ディスクで扱うファイル数は，ディレクトリの大きさとクラスタ数から決まります．

ディスクタイプ	ディレクトリの大きさ（セクタ数）	ファイル数
3.5インチおよび5インチ両面倍密度倍トラックフロッピィディスク	12	159
3.5インチ，5インチ高密度および8インチ両面倍密度フロッピィディスク	22	151
固定ディスク（5MB）	99	603
固定ディスク（10MB）	99	1231
固定ディスク（20MB）	231	2443

(2)　ディレクトリの形式

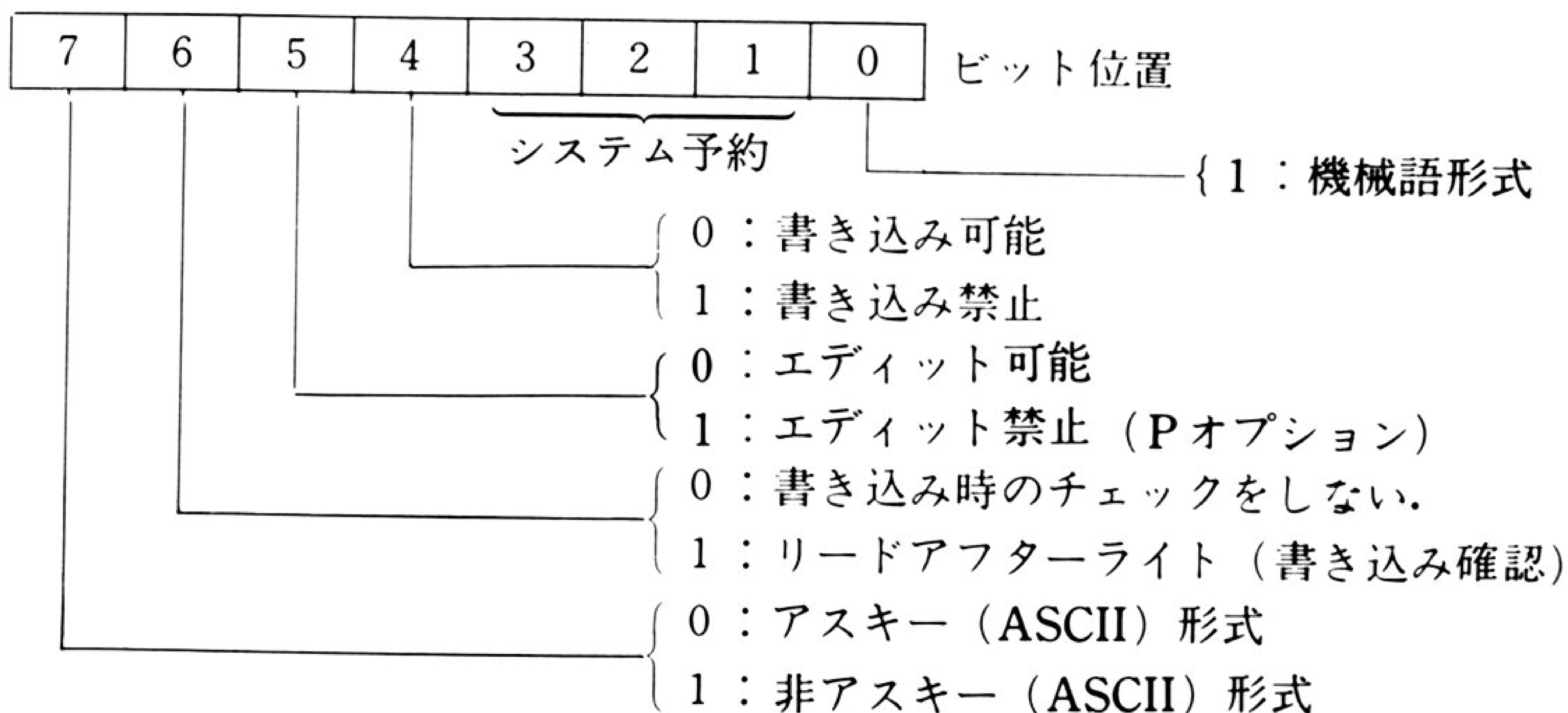
ファイル毎にもつ16バイトの制御情報は，次のような内容をもっています．ただし，ファイルが格納されている先頭のクラスタ番号を示すバイト数が，固定ディスクでは2バイトを使用し，

他のディスクでは１バイトを使用しています．

ファイル名の先頭バイトの内容がFF（16進数）のとき，そのディレクトリは未使用であることを示します．０のとき，削除（Kill）されたファイルであることを示します．

フロッピィディスク		固定ディスク		内　容
位　置（バイト）	（バイト）大きさ	位　置（バイト）	（バイト）大きさ	
0～5	6	0～5	6	ファイル名
6～8	3	6～8	3	拡張子
9	1	9	1	属　性
10	1	10～11	2	そのファイルが格納されている先頭のクラスタ番号
11～15	5	12	1	システム予約
		13～15	3	ユーザ識別名

属性を表わす１バイトは次のような意味を持っています．



12.5 FAT

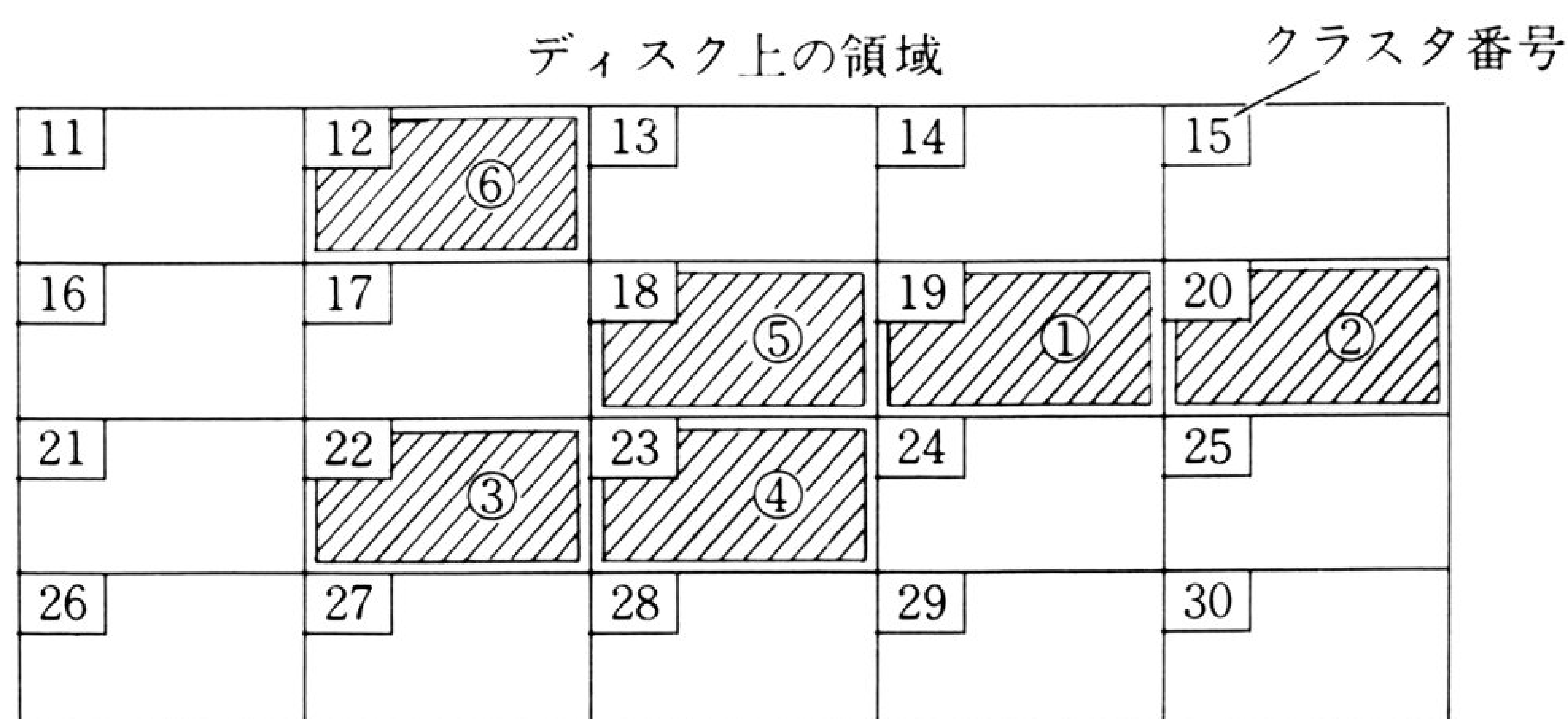
(1) FATの役割

FAT（ファイル領域管理テーブル）は，ファイルが使用するディスク領域を管理させるための表です．ディスク領域は，クラスタを単位に管理されます．各クラスタの使用状況を表わすために，クラスタごとに1バイト，または2バイトの制御情報をもちます．固定ディスクは2バイト，他のディスクは1バイトを使用します．これを1エントリとしたディスク上の全クラスタに対応した制御情報の集りがFATです．

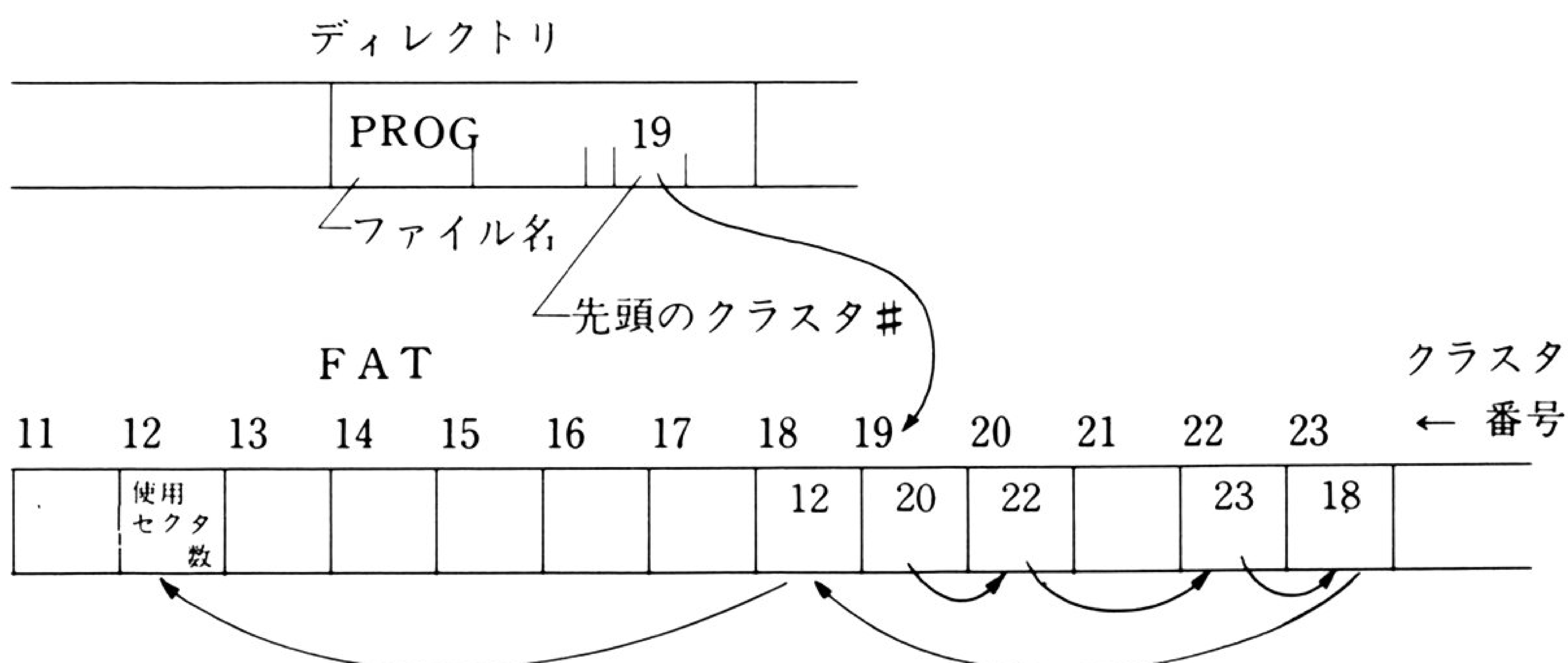
FATはファイルが使用しているディスク領域をエントリ間のチェイン関係により表わします．

例を通してFATの構造を説明します．

下図は，ディスク上の領域を示しています．ファイル“PROG”がクラスタ番号19を最初の領域とし，つづいて，20，22，23，18，最終の領域としてクラスタ番号12の領域を使用していることを示しています．



この関係をFATは次のように管理します．また，ディレクトリには先頭のクラスタ番号が示されています．



使用中の後続のクラスタをもつクラスタに対応するエントリには後続のクラスタ番号が格納されます．

使用中の最終のクラスタになるクラスタのエントリには，そのクラスタ内の実際に使用されているセクタ数を表わす数値が示されます．

この例では示していませんが，その他にシステムで予約済の領域にあたるクラスタに対応するエントリには16進数でFEが格納され，未使用のクラスタに対応するエントリには16進数でFFが格納されます．

(2)　FATの内容

FATの各エントリは対応するクラスタの状態によって次のような値を持ちます．固定ディスクの場合は2バイト，他は1バイト情報です．

クラスタの状態	対応するFATエントリ（1または2バイトの内容）	対応するFATエントリの具体的な内容				
		3.5インチおよび5インチ両面倍密度倍トラック	3.5インチ，5インチ高密度および8インチ	固定ディスク（5MB）	固定ディスク（10MB）	固定ディスク（20MB）
後続したクラスタをもつクラスタ（使用中の状態）	後続するクラスタのクラスタ番号を示す	0～9F	0～99	0～263	0～4D7	0～99B
最終のクラスタ（使用中の状態）	このクラスタで使用済のセクタ数を示す	C0～D0	C0～DA	4000～4021	4000～4021	4000～4021
システムが予約済のクラスタ DISK CODE, IPL，ディレクトリ，FAT, ID等を含むクラスタ		FE	FE	FFFE	FFFE	FFFE
エラークラスタ（使用不可のセクタをもつクラスタ）				FFFD	FFFD	FFFD
未使用クラスタ		FF	FF	FFFF	FFFF	FFFF

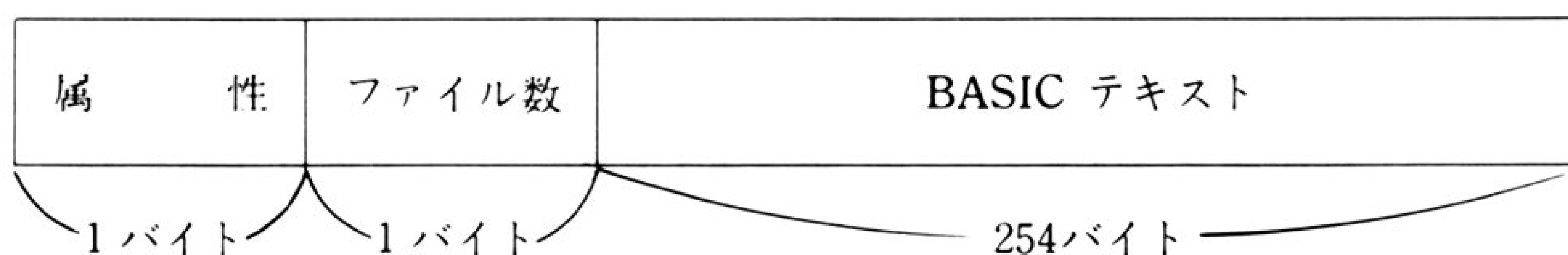
また，FATの大きさはディスクの種類によって異なります．固定ディスクはクラスタの総数の2倍のバイト数です．他のディスクはクラスタ総数と同じバイト数ですが，この場合は同じ内容をリカバリのために3重書きしますので，さらに3倍した値の大きさを使用しています．これは有効バイト数ですが，FATのための割当てトラック数は十分大きくとっています．

FATは，その装置に対する最初のアクセスでメモリ上に読み込まれ，FATを更新する機会に，必要に応じて媒体へ書きもどされます．

12.6 ID

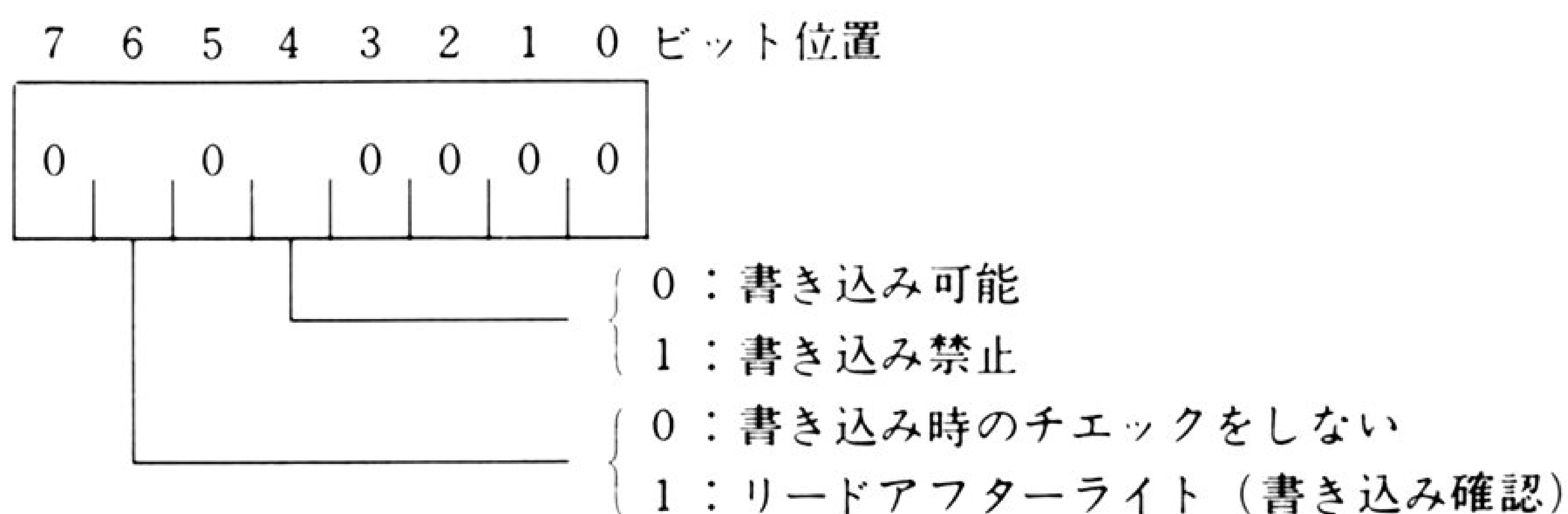
(1) IDの形式

ディスクには，ディスク媒体を識別するために１セクタ分の領域が確保されています．この領域をIDとよびます．IDは次のような形式をしています．



・属性

ディスク媒体の属性を示します．



set〈ドライブ番号〉,〈“属性文字”〉

で“属性文字”が“P”のとき，書き込み禁止，

“R”のとき，リードアフターライト（書き込み確認）

を指示し，それぞれ，4ビット目，6ビット目がセットされます．

・ファイル数（システムディスク用）

同時にオープンされるファイル数を指定するエリアです．

通常のシステムディスクでは255（16進数でff）が指定されています．

この〈ファイル数〉に0から15までの数値を指定すると，BASICをスタートさせたとき，

How many files（0－15）？

を表示せずに，自動的にそのファイル数が設定されます．

・BASICテキスト（システムディスク用）

オートスタート時に自動的に最初に実行されるBASICのステートメントを書き込んでおくエリアです．すなわち，オートスタート時にはBASICの最初の実行は，このテキストからはじまります．実行が終了するとBASICの入力可能状態になります．

(2) IDの例

属性　ファイル数　BASICテキスト

0 0	0 5	7 2	7 5	6 E	2 2	4 1	0 0 ……………… 0 0
		r	u	n	"	A	

属性：指定なし

ファイル数：5

BASICテキスト：run "A

(3) IDへの書き込み方

IDへの書き込みのための特別なコマンドはありません.

ユーティリティ「setinf.n88」(IDセクタの書き換え)を使います.

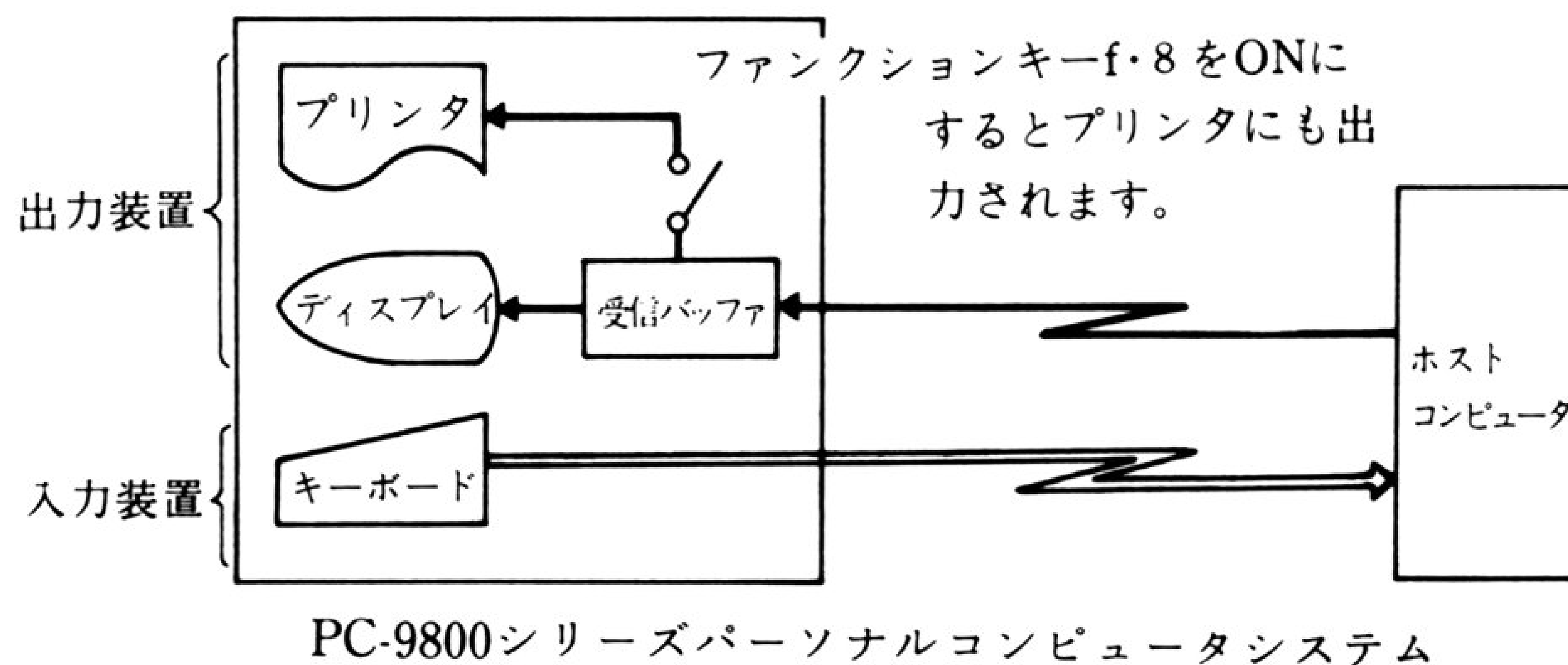
第13章
ターミナルモード

13.1 端末装置の仕様

PC-9800シリーズパーソナルコンピュータシステムを他のコンピュータの端末装置として使用するためにターミナルモードが用意されています．どのような仕様の端末装置になるのかを，ホストコンピュータとのデータ送受信の流れを通して説明します．

(1) **端末装置としてのデータフロー**

(a) 全二重モードの端末装置になった場合



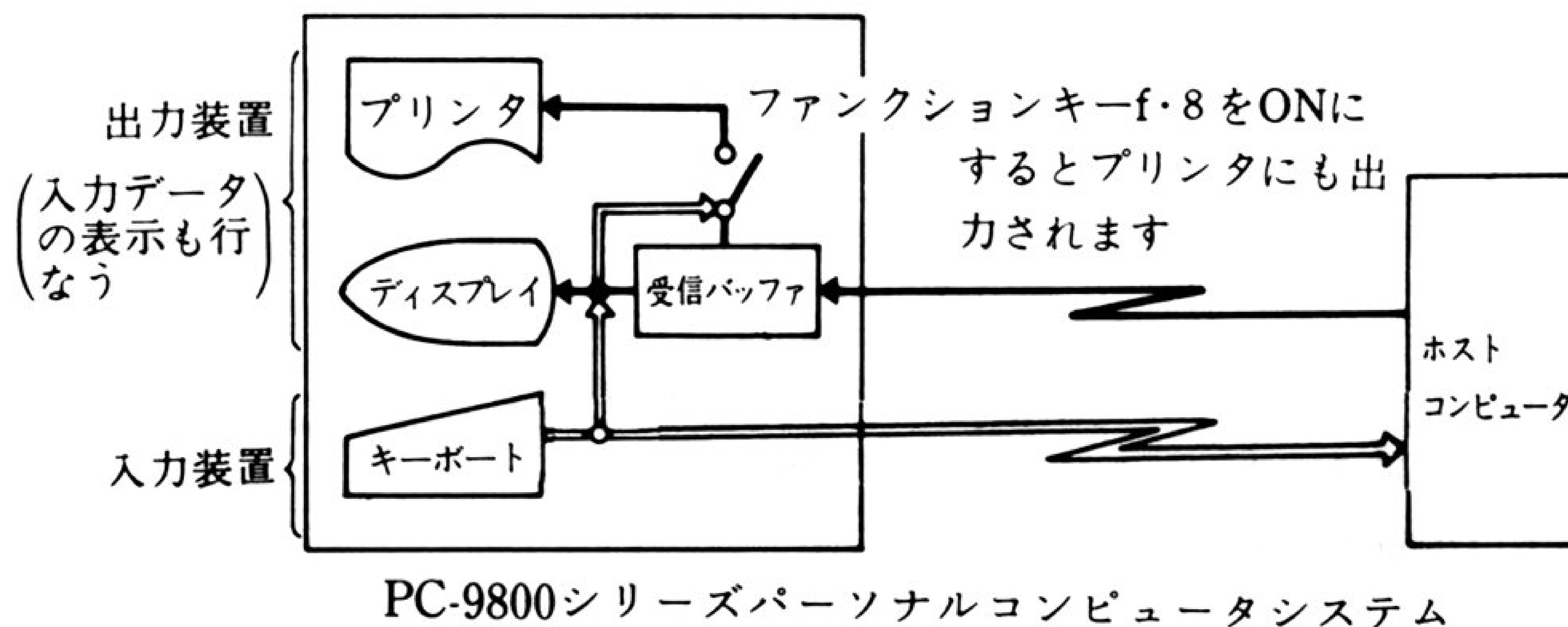
PC-9800シリーズパーソナルコンピュータシステム

① キーインされたデータは，ただちに回線に送出されます．また，ディスプレイ，プリンタへは表示・出力されません．

② 受信データはディスプレイに表示されます．ファンクションキーf・8がONならば同時にプリンタに出力されます．

③ ホストコンピュータからのデータ受信と送信は，独立に行われます．

(b) 半二重モードの端末装置になった場合



PC-9800シリーズパーソナルコンピュータシステム

① キーインされたデータはいったんディスプレイに表示されます．プリンタが使用できるときは，プリンタにも出力されます（プリンタが接続されており，ファンクションキー[f･8]がONの場合）

② キーインされたデータの送信は，[⏎]キー（または[CTRL]＋M）を押下することにより，行われます．

③ 送信が終ると，受信データをディスプレイへ表示します．プリンタが使用できるときは，プリンタにも出力されます．

(2) 端末装置の諸元

次の表は端末装置の諸元を示しています．固定的なものと，選択可能なものとがあります．これらの選択可能なものを定める方法は，後で説明します．

項番	諸　元	仕様および選択可能範囲
1	伝送制御手順	無手順
2	通信方式	全二重/半二重（回線上は全二重）
3	フロー制御（Xパラメータ）	受信データのフロー制御を行う/行わない
4	データビット長	7ビット/8ビット
5	ストップビット長	1ビット/1.5ビット/2ビット
6	パリティチェック，パリティ種類	偶数/奇数/なし
7	ボーレート	75/150/300/600/1200/2400/4800/9600
8	コード	JIS7 (ASCII), JIS8, JIS C6226（日本語）
9	使用モデム	全二重
10	同期方式	非同期
11	インタフェース	RS-232C
12	データビット長7の時のシフトコード制御（Sパラメータ）	7ビットモードでカナの送受信ができます/できません
13	DELコード処理	BSコードに変換/NULコードに変換
14	RETURNキー処理	C_Rコード送信/C_R+L_Fコード送信
15	受信C_Rコード処理	C_Rコード単独で，復帰＋改行動作/C_R+L_F連続で，復帰＋改行動作
16	日本語シフトコード	((KI)=ESC･K$(1B4B)_{16}$, (KO)=ESC･H$(1B48)_{16}$)/((KI)=SB･p$(1A70)_{16}$, (KO)=SB･q$(1A71)_{16}$)
17	ブレーク信号処理	送信：STOPキーの押下により送信されます 受信：ホストコンピュータからのブレーク信号は受け付けません
18	日本語コード（項番8参照）	JIS C 6226コード

表の見方 (1)　/はいずれかを選択できることを示します．

(2)　7ビットモードでのカナの送受信は，シフトコードSO（16進数0E）があると，それ以後のデータはカナとみなし，シフトコードSI（16進数0F）がくるとそれ以後のデータを英数字とみなします．初期値はSIとみなします．

注　意 表記のボーレートは，RS-232Cハードウエア（回路）上の性能であり，ソフトウエア的な要因によっては，記述の速度による通信が出来ない場合もあります．

〈ソフトウエアの要因〉

Ⅰ．一度に転送するデータ量

Ⅱ．周辺装置に対するI/Oの有無

Ⅲ．アプリケーションプログラム内での通信データの入出力間隔

以上があります．

これら要因による，通信データの抜けを防止する為，フロー制御という手法が用いられます．（フロー制御については次項を参照下さい）

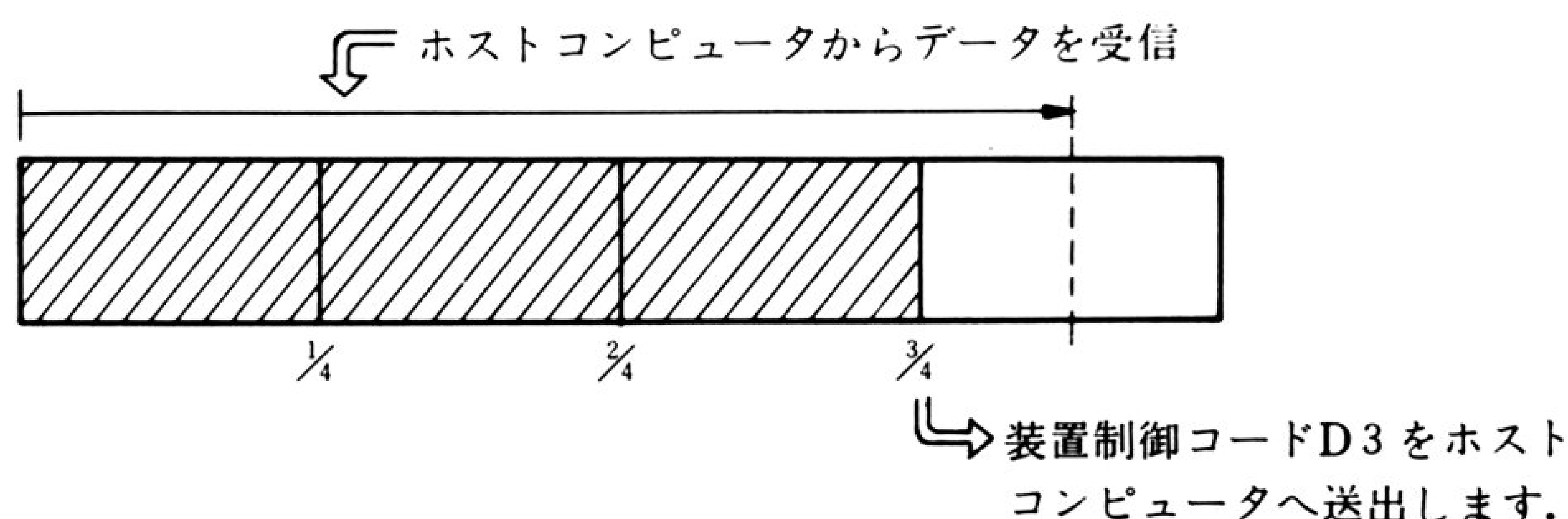
(3)　フロー制御と装置制御コードD1/D3処理

ターミナルモードは受信用バッファの使用度による開閉制御を行う機能をもっています．これを受信データのフロー制御とよびます．

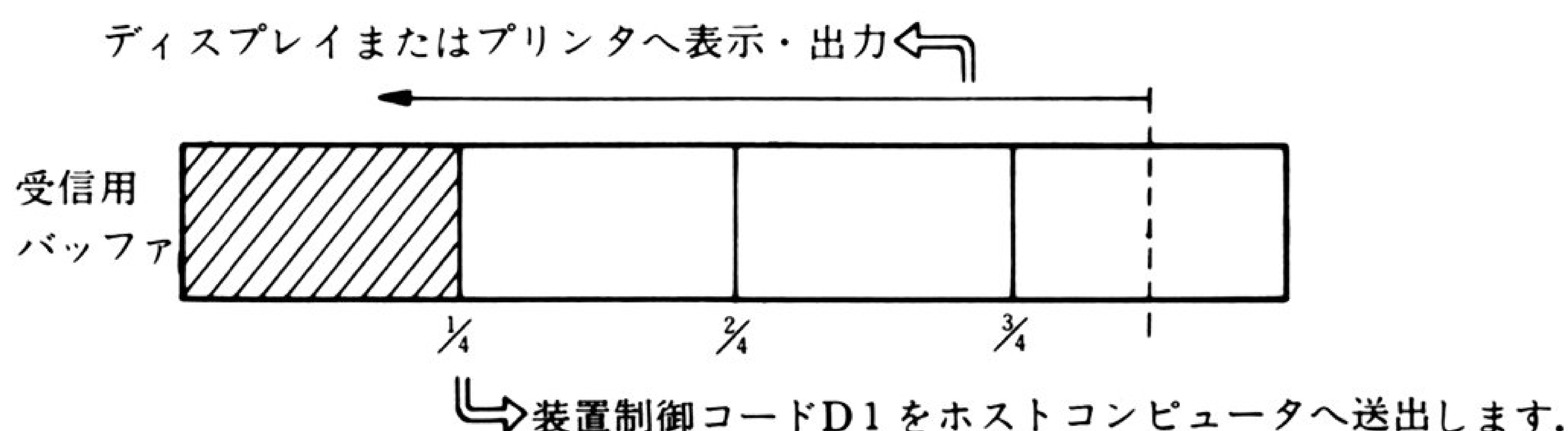
(a)　受信データのフロー制御（(2)の表・項番3）

受信用バッファ中のデータが，バッファ域の3/4を超えるとホストコンピュータへ装置制御のコードD3（16進数13）を送信します．

これにより，ホストコンピュータからの送信を一時停止するように要求します．



また，一度，装置制御コードD3を送信した後，受信用バッファ中のデータ処理が進み，データ量がバッファ域の1/4より少なくなると，装置制御コードD1（16進数11）を送信し，ホストコンピュータからの送信再開を要求します．



(b) ホストコンピュータからの装置制御コードD 1/D 3受信時の処理

ホストコンピュータが，端末装置から送信してくるデータの受信を停止または再開するために装置制御コードD 3またはD 1をPC-9800シリーズパーソナルコンピュータシステムに送信してきた場合，どのような処理が行われるか説明します．

(イ) D 3受信時の処理

① 全二重モードの場合

回線への送出を直ちに停止します．その後にキーインされたデータは無効とします．

② 半二重モードの場合

すでに，キーイン中のときは，[↵]キー押下で1行送信後，以後の送信を停止します．その後にキーインされたデータは無効とします．

(ロ) D 1受信時の処理

全二重モード，半二重モードいずれの場合にも再び送信可能状態になります．

なお，何らかの原因で，ホストコンピュータがD 3を送信し，その後のD 1の送信がない場合，[SHIFT]＋[STOP]キーを押下することにより，強制的に送信可能状態に戻すことが可能です．

(4) **スクロール用メモリについて**

この端末装置は，受信データまたはキーボードから入力したデータをディスプレイ装置に出力しますが，スクロールエリアがいっぱいになる（カーソルが最終行右端にくる）と自動的にスクロールアップします．スクロールアップによってスクロールエリアからオーバフローした行の内容は，スクロール用メモリにたくわえられます．また，これをファンクションキーによってプリンタへ出力することができます．

(5) **日本語コードの送受信について**

(a) 日本語の入力

日本語データを送信するためにキーボードから日本語コードを入力する方法は次のとおりです．

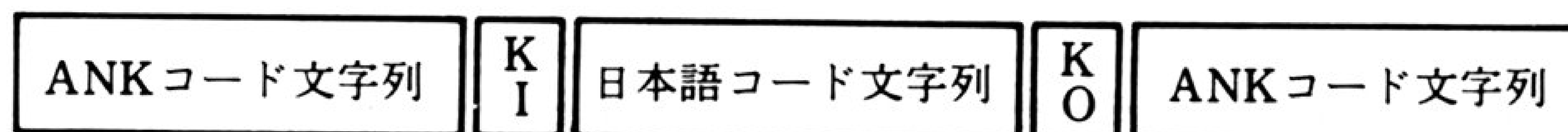
・[CTRL]＋[XFER]キー

日本語データ入力開始/入力終了のためのキーとして使用します．開始/終了は押下動作の順に交互に変更されます．(開始→終了→開始→…)

・日本語コードの入力

16進数表記（0～9，a～f）4桁のJISコード入力で入力します．JISコードについてはハードウエアマニュアルのコード表を参照してください．

(b) 回線上では，日本語コードと英数カナ（ANK）コードは，日本語シフトコードKI/KOコードの挿入によって識別可能です．



送／受信方向(順)→

(c) 日本語シフトコードKI/KOの指定

2種類のシフトコードがあります．メモリスイッチ，またはterm文で指定します．

(d)　送信の場合

・Sパラメータ有効時，日本語コード文字列は，SI側（ANKコード）文字列の一部として送信されます．

（例）

SI	KI	味噌	KO	SO	ト	SI	KI	少	KO	SO	シノ	SI	KI	野菜	KO	SO	ヲタベ	CL RF

・日本語コード文字列中に制御符号（CRなど）を挿入するときは直前にKOを挿入します．

・日本語1字を表わす第1バイトと第2バイトの間に制御符号（$(00)_h$～$(1F)_h$のコード）が入ることはありません．

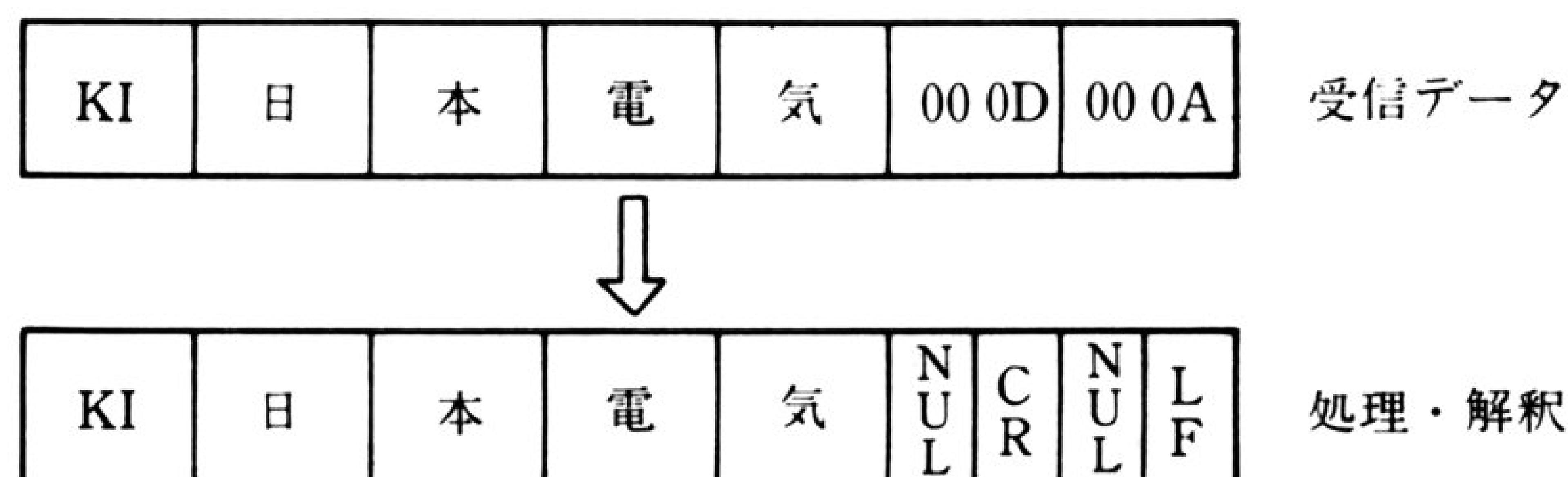
(e)　受信の場合

・日本語コードの第1バイトと第2バイトの間に他コードを挿入してはいけません．

・1つ前の文字の第2バイトと次の第1バイトの間には1バイト系の制御符号が入っていてもかまいません．

第1バイトが日本語コードでないときは，1バイト系の制御符号として処理します．

（例）



13.2　端末装置仕様の設定方法

(1)　2つの設定方法

PC-9800シリーズパーソナルコンピュータシステムを端末装置として使用する場合に，端末装置としての仕様設定には次の2通りの方法があります．

①　メモリスイッチによる設定

②　term文による設定

(a)　メモリスイッチを使用する場合

これは端末装置の仕様をある程度固定化してしまう場合に便利です．システム立ち上げ後の変更はBASICモードでterm文を実行することによって可能です．term文により変更された値が有効なのはこの命令の実行によってターミナルモードになっている間です．

メモリスイッチにセットされている条件は次のセットがあるまでシステムの稼動，停止にかかわらず有効です．変更はセットと同時に有効になります．メモリスイッチのセットは「第21章メモリスイッチ」に説明してあります．

(b) term 文による設定

term 文による設定は，端末装置の仕様を柔軟に設定し，多様な活動を期待する場合に便利です．接続するホストコンピュータ，アプリケーションの相違によって，最も適した仕様をその都度設定することができます．term 文によって設定された仕様が有効なのは，このターミナルモード中のみです．

(2) term 文による方法

① ② ③ ④ ⑤ ⑥ ⑦ ⑧

$$\text{term}''[\text{COM}:]\ [\begin{Bmatrix}E\\O\\N\\\triangle\end{Bmatrix}\ [\begin{Bmatrix}7\\8\\\triangle\end{Bmatrix}\ [\begin{Bmatrix}1\\2\\3\\\triangle\end{Bmatrix}\ [\begin{Bmatrix}X\\N\\\triangle\end{Bmatrix}\ [\begin{Bmatrix}S\\N\\\triangle\end{Bmatrix}\ [\begin{Bmatrix}B\\N\\\triangle\end{Bmatrix}\ [\begin{Bmatrix}C\\L\\\triangle\end{Bmatrix}\ [\begin{Bmatrix}C\\L\\\triangle\end{Bmatrix}$$

⑨ ⑩ ⑪

$$[\begin{Bmatrix}P\\I\\\triangle\end{Bmatrix}]]]]]]]]]''\ [,\ [\begin{Bmatrix}F\\H\end{Bmatrix}]\ [,\ \text{配列エリア長}]]$$

次にパラメータの内容とそれに対応するメモリスイッチの位置および状態を示します．

項番	パラメータの種類	パラメータの値	意　　味	対応するメモリスイッチ
1	パリティ	E	偶数パリティでチェックします	$(54)_{SW1}=11$
		O	奇数パリティでチェックします	$(54)_{SW1}=01$
		N	パリティチェックをしません	$(4)_{SW1}=0$
		ブランク，省略	メモリスイッチの指定値	
2	データビット長	7	データのビット長は 7 ビット	$(32)_{SW1}=10$
		8	データのビット長は 8 ビット	$(32)_{SW1}=11$
		ブランク，省略	メモリスイッチの指定値	
3	ストップビット長	1	ストップビットの長さが 1 ビット	$(76)_{SW1}=01$
		2	〃　1.5ビット	$(76)_{SW1}=10$
		3	〃　2 ビット	$(76)_{SW1}=11$
		ブランク，省略	メモリスイッチの指定値	
4	X パラメータ	X	受信データのフロー制御を行う	$(0)_{SW1}=1$
		N	〃　行わない	$(0)_{SW1}=0$
		ブランク，省略	メモリスイッチの指定値	

項番	パラメータの種類	パラメータの値	意味	対応するメモリスイッチ
5	Sパラメータ	S N ブランク，省略	データビット長7ビットでカナを扱う 〃 扱わない メモリスイッチの指定値	$(7)_{sw2}=1$ $(7)_{sw2}=0$
6	DELコード受信処理	B N ブランク，省略	$BS(08)_{16}$コードとして扱う $NUL(00)_{16}$コードとして扱う メモリスイッチの指定値	$(7)_{sw3}=0$ $(7)_{sw3}=1$
7	RETURNキー押下時の送信処理	C L ブランク，省略	C_Rコード単位で復帰＋改行動作 C_R+L_F連続で復帰＋改行動作 メモリスイッチの指定値	$(6)_{sw2}=0$ $(6)_{sw2}=1$
8	C_R受信処理	C L ブランク，省略	C_Rコード単独で復帰＋改行動作 C_R+L_F連続で復帰＋改行動作 メモリスイッチの指定値	$(5)_{sw2}=0$ $(5)_{sw2}=1$
9	日本語シフトコード指定	P I ブランク，省略	(KI)＝ESC·K (KO)＝ESC·H (KI)＝S_B·p (KO)＝S_B·q メモリスイッチの指定値	$(4)_{sw2}=0$ $(4)_{sw2}=1$
10	通信方式	F H 省略	全二重方式 半二重方式 メモリスイッチの指定値	$(1)_{sw1}=0$ $(1)_{sw1}=1$
11	配列エリア長	〈数値〉 省略	BASICプロトコルの実行時に使用する配列エリアの大きさを指定 省略時は1024バイト	指定しない

表の見方「メモリスイッチの状態」の項の見方

・たとえば，$(54)_{sw1}=11$はメモリスイッチSW1の5ビット・4ビットがそれぞれ1·1にセットされていることを示します．これは，パリティチェックを行い，かつ偶数パリティを使うことを意味しています．

・もう一つの例，$(76)_{sw1}=10$はメモリスイッチSW1の7ビット・6ビットがそれぞれ1·0にセットされていることを示します．これは，ストップビット長が1.5ビットであることを意味します．

注　意　配列エリア全体の大きさからBASICで実際に使用していた配列エリアの大きさとBASICプロトコル用の配列エリアの大きさを差引いた領域がターミナルモード時の通信用バッファとスクロールエリアの大きさになります．

例 term "com：E72X△△△△P", H

偶数パリティ，データビット長７ビット，ストップビット長1.5ビット，受信用バッファのフロー制御を行う，日本語シフトコードにESC・K/ESC・HをKI/KOとして用いる，半２重モード，BASICプロトコルの実行時の数値型配列エリアの大きさは1024バイトの端末装置として機能します．△はブランクを示します．

例 term "com：E"

偶数パリティで，他はメモリスイッチ通りにセットされます．

例 ここでは，実際に使用した例を示します．

ただし，それぞれのホストシステムに接続できる端末装置の仕様はここで示した例だけではありません．また，ホストの条件によってはこの例が必ずしも適切ではないかもしれません．

① PC-VANの端末として使用した例

term "COM：N81NNBCLP"

② VAX780UnixのTSS端末として使用した例

term "COM：E71XNBCLP", F

(3) **メモリスイッチによる方法**

・メモリスイッチのセットの仕方

詳しくは「第21章メモリスイッチ」を参照してください．機械語モニタモードによって，直接メモリスイッチのビットの内容を０，または１にセットします．s（セット・メモリ）コマンドを使用します．

・メモリスイッチの位置と機能

SW1

ビット位置	7	6	5	4	3	2	1	0
機　能	ストップビット長		パリティ種類	パリティチェック	データビット長		通信方式	Xパラメータ
0：OFF	01：1ビット(1) 10：1.5ビット(2) 11：2ビット(3)		奇数(O)	なし(N)	10:7ビット長(7)		全二重(F)	無効(N)
1：ON			偶数(E)	あり	11:8ビット長(8)		半二重(H)	有効(X)

システム既定値(48)₁₆

SW2

ビット位置	7	6	5	4	3	2	1	0
機　能	Sパラメータ	RETURNキー送信コード	C_R受信処理	日本語シフトコード	ボーレート			
0：OFF	SI·SO なし(N)	C_R(C)	C_R 復帰＋改行(C)	(KI)＝ESC·K (KO)＝ESC·H (P)	0000:無効 0001:75ボー 0010:150ボー	0011:300ボー 0100:600ボー 0101:1200ボー	0110:2400ボー 0111:4800ボー 1000:9600ボー	
1：ON	SI·SO あり(S)	C_R+L_F (L)	C_R+L_F 復帰＋改行(L)	(KI)＝S_B·p (KO)＝S_B·q (I)				

システム既定値$(05)_{16}$

SW3

ビット位置	7	6	5	4	3	2	1	0
機　能	DEL コード受信処理	他の目的に使用しています						
0：OFF	BSコード(B)							
1：ON	NULLコード(N)							

システム既定値$(00)_{16}$

表の見方　1.（　）内の太文字は，該当する**term**文のパラメータの値を示しています．

2．ボーレートは，メモリスイッチによってのみ指定可能です．

3．複数ビットを使用する項は，ビットの並びと2進の桁数とが対応しています．たとえば，SW1のストップビット長の項「10：1.5ビット(2)」は，ビット位置7が1，ビット位置6が0を示しています．

例　Xパラメータ有効，全二重，7ビット長，偶数パリティ，ストップビット長1.5，ボーレート1200ボー，日本語シフトコードS_B・p/S_B・q，C_Rコード受信で，復帰＋改行動作，RETURNキーで$C_R(0D)_{16}$コード送信，シフトコードなし，DELコード無視（SW3の他の目的で使用する項すべて0）

SW1＝10111001＝&HB9　SW2＝00010101＝&H15

SW3＝10000000＝&H80

(4)　ボーレートと同期の設定

(a)　ボーレート（伝送速度）の設定

ボーレート（伝送速度）はホストコンピュータとの調整を行った後設定してください．前項で少し説明していますが，この設定もメモリスイッチで行います．システム既定値は1200ボーにセットされています．

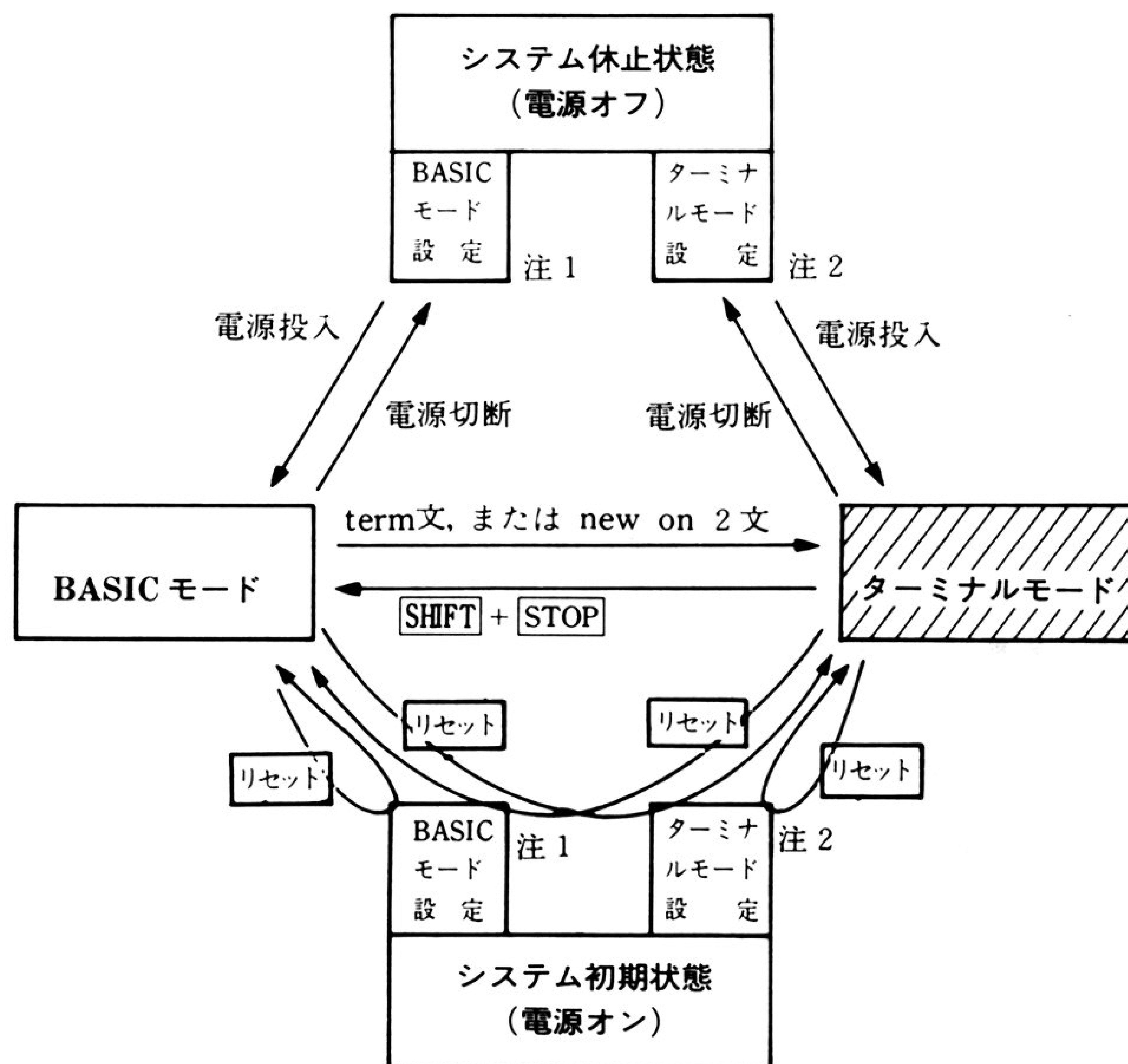
(b)　同期クロックの設定

一般には，RS-232CのボーレートをカウントするためのクロックをPC-9800シリーズパーソナルコンピュータのタイマからとる（内部同期）か，モデムからとる（外部同期）かを選択することができます．しかし，ターミナルモードでは内部同期だけしか使用できません．この選択にはディップスイッチSW1の5，6を使用しますがターミナルモードにおいては常にOFFにしておきます．

13.3　ターミナルモードへ（から）の切り換え

ターミナルモード以外のシステム状態から，ターミナルモードへ切り換える方法，また，ターミナルモードから他のシステム状態に切り換える方法を説明します．

ターミナルモードからみたシステム遷移図は次のようになります．

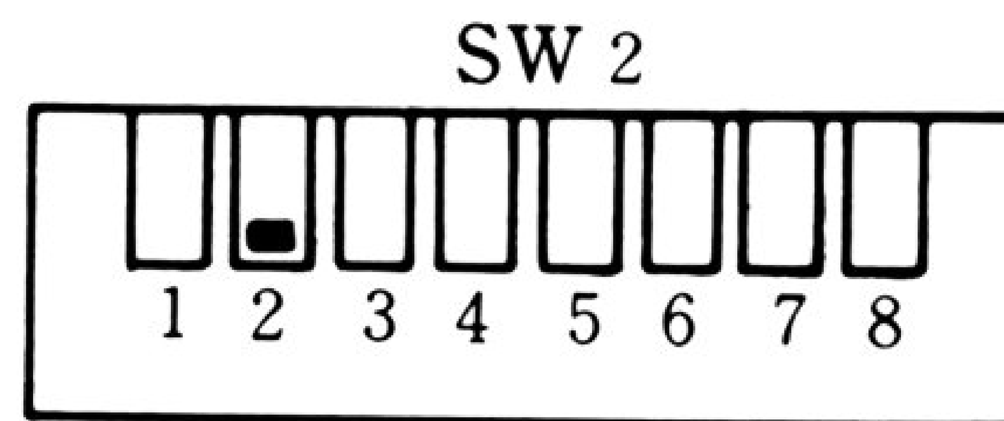


(1) ターミナルモードとしてシステムを立ち上げる方法

・DIP SWによる方法
・オートスタート用BASICテキスト（IDセクタ上にある）を使う方法
の2通りあります．

(a) ディップ・スイッチでターミナルモードに設定

DIP SW 2の2は，システムの実行モードを設定するスイッチです．これを「ON」にするとターミナルモードになります．この場合には，システム立ち上げ時のモードは常にターミナルモードになります．



遷移図の注2は，上のDIP SW 2の2が「ON」になっている状態です．注1は「OFF」の場合です．

(b) オートスタート用BASICテキストを使う方法

システムディスクのIDセクタ256バイトには，オートスタート時の情報を書き込むことができます．オートスタートというのは，システム立ち上げ時，自動的に指定した処理を行う動作です．この動作の中で，**term**文または**new on** 2文を実行させて，ただちにターミナルモードにしてしまうことができます．システムユーティリティ **setinf.n88**を使ってオートスタート情報を設定してください．**new on 2**文は，DIP SW 2の2をBASICで一時的にセットするためのものです．

```
new on 2
```

でターミナルモードになります．これをIDセクタのBASICテキストに書き込んでおきます．すると，オートスタートによって，ターミナルモードになります．

遷移図の注2は，IDセクタのBASICテキストを**term**文，または**new on** 2文を書き込んでおくことによっても設定できます．

ただし，この場合，制御上はいったんBASICモードになっているのですが，ユーザに直接みえないので，注2の状態にしておきます．

(2) BASICモード，ターミナルモードの切り換え方法

(a) BASICモードからターミナルモードへ

BASICモードで，**term**文，または**new on** 2文を実行すると，ターミナルモードになります．

(b) ターミナルモードからBASICモードへ

ターミナルモードで，SHIFT＋STOPを押下することによってBASICモードになります．ターミナルモードの動作中にラインバッファオーバフローが発生していたら，BASICモードになったときに“Line buffer overflow”のメッセージが出力されます．

13.4 端末装置の使い方

具体的に端末装置として操作していくために知っておかなくてはならない事柄を説明します．

(1) ファンクションキー

ターミナルモードにおいても，10種類のファンクションキーが使用できます．ディスプレイにもファンクションキーをBASICモードと同じように表示できます．もちろん，console文でファンクションキー表示スイッチを「ON」にしておく必要があります．

BASICモードと異なっているところを説明します．

(a) [f･1] から [f･5] までの機能

ファンクションキーの内容はターミナルモードになる直前の状態が引継がれます．

(b) [f･6] から [f･10] までの機能

① ファンクションキーの機能定義

[f･6]：制御コードの扱い方を選択するために使用します．
制御コードを「制御コード」として処理する場合を“N-LTRL”とよび，制御コードを文字として処理し，「制御コード」としては処理しない場合を“LTRL”とよびます．この場合，C_R，L_Fだけは「制御コード」として機能します．また，画面への文字表示はすべての制御コードについて行います．

[f･7]：通信方式を定義するために使用します．
半二重通信方式を“HALF”とよび，全二重通信方式を“FULL”とよびます．

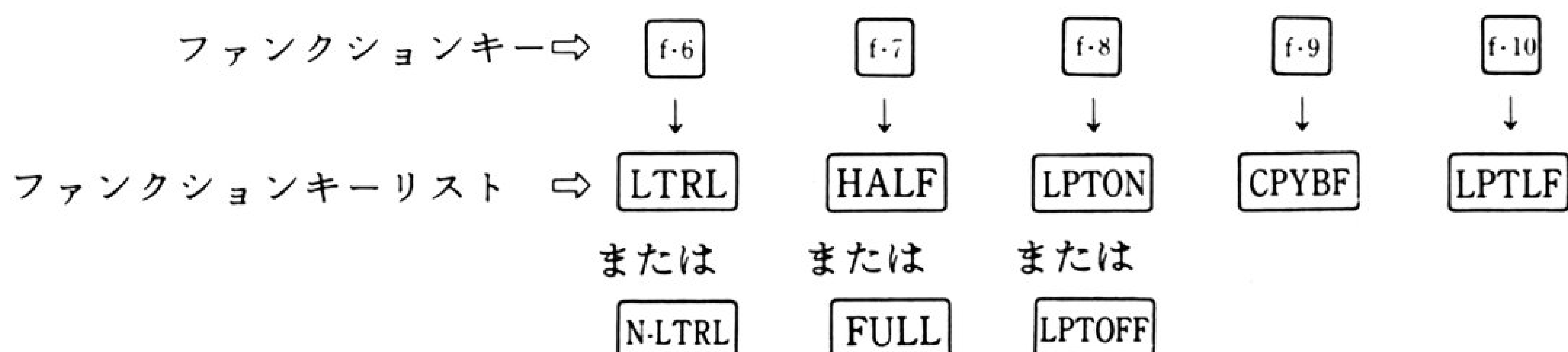
[f･8]：画面に表示する内容をプリンタにも出力するかどうかを選択するために使用します．
プリンタに出力しない場合を“LPT OFF”とよび，出力する場合を“LPT ON”とよびます．

[f･9]：画面スクロール用メモリの内容をすべてプリンタに出力することを要求するために使用し，この機能を“CPY BF“とよびます．
“LPT ON”のとき有効です．

[f･10]：プリンタを1行改行するために使用し，この機能を“LPT LF”とよびます．“LPT ON”のとき有効です．

② ファンクションキーリストの表示状態

テキスト画面最下位行に表示されるファンクションキーリストは，[f･6] から [f･8] まではキーを押下することによって機能する機能名が表示されています．[f･9]，[f･10] は機能名が表示されています．

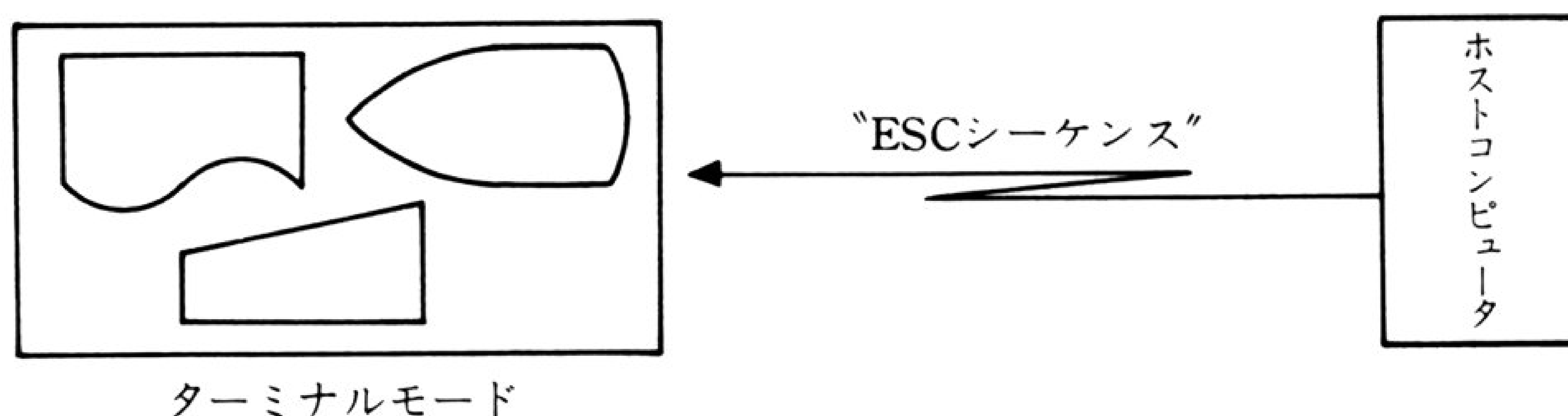


注　意　f･6 ～ f･8 についてはファンクションキーの押下によって機能している現在の状態と，ファンクションキーリストに表示されている機能名とは逆の関係になっています．ファンクションキーを押下すると，現在表示されている機能名の機能状態になり，ファンクションキーリストには逆の機能名が表示されます．

(2)　スペシャルESCシーケンス

ホストコンピュータから端末装置を制御するためにシステム間の取り決め（プロトコル）があります．この取り決めは1つの手順になっています．

この手順は，ESC（エスケープ）(1B)ₕコードで始まる一連のキャラクタストリング（これをスペシャルESCシーケンスとよびます）で，ホストコンピュータからこれらの情報が送信されると，端末装置はこれを受信し，各ESCシーケンスに定義されている機能を実行します．これによって，端末装置側の状態を設定・変更できます．



定義されているスペシャルESCシーケンスとその機能・動作は次のようになっています．

① ESC@　　ディスプレイへの出力は，以後，プリンタへも出力されます．

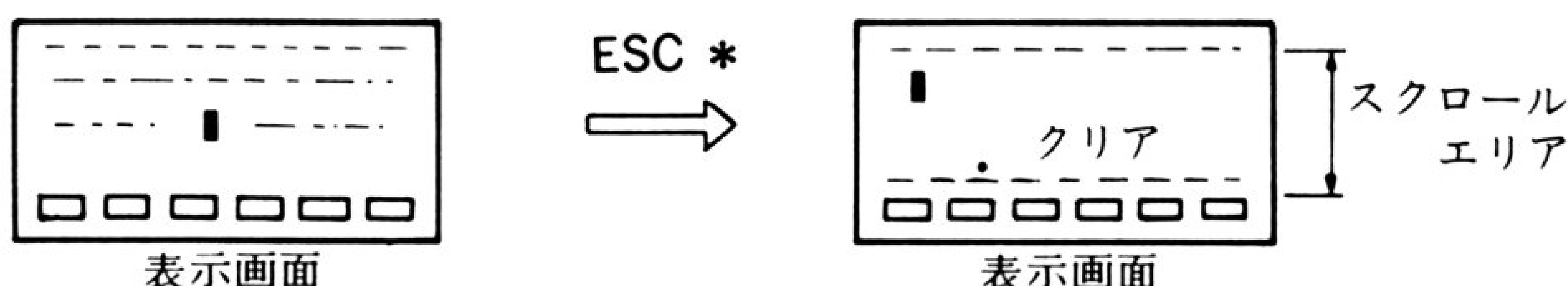
② ESC A　　ディスプレイへの出力は，以後，プリンタへは出力しません．①の状態を解消します．

③ ESC"　　キー入力を有効にします．

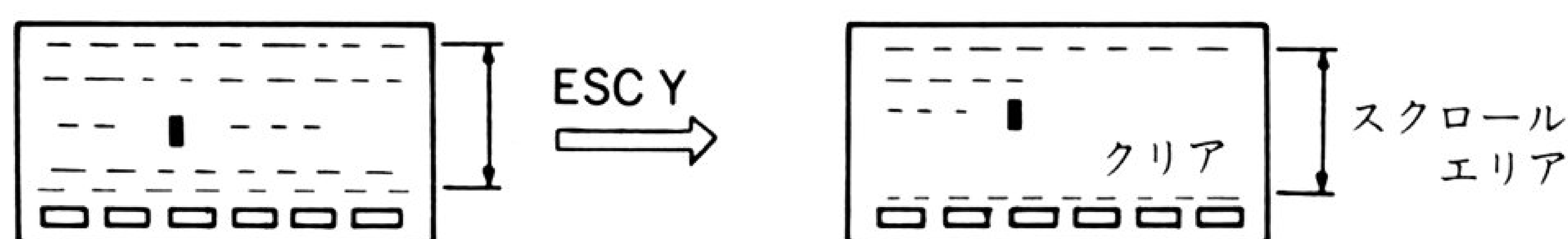
④ ESC♯　　キー入力を無効にします．③の状態を解消します．以後，ESC"を受信するまでキー入力されたデータは捨てられます．

⑤ ESC！　　ホストコンピュータへIDコードを送信します．IDコードは"NEC00000010 CR [LF]"です．ディスプレイ/プリンタは，何も出力しません．

⑥ ESC＊　　表示画面（ディスプレイ画面）をクリアします．クリア後，カーソルは表示画面（ディスプレイ画面）の左上角にセットされます．



⑦ ESC Y　　現在のカーソル位置から，スクロールエリアの最下位行までをクリアします．カーソル位置は変化しません．



⑧ ESC△　ターミナルの状態をターミナルモードになった直後の状態にもどします（ただし，画面はクリアされません）.

⑨ ESC＝ij　カーソルの位置を移動させます．座標はテキスト画面です．移動する位置を（列，行）＝（p, q）とすると，

$i = p + (20)_h$

$j = q + (20)_h$

という関係をもちます，i, jは16進数２桁の数値に対応する文字です．$(20)_h$のバイアスは制御コードを避けるためです．

（例）カーソルの位置を（３，５）に移動させるためには，

$i = 3 + (20)_h = (23)_h \rightarrow \#$

$j = 5 + (20)_h = (25)_h \rightarrow \%$

すなわち，ESC＝＃%と指定することになります．

⑩ ESC T　現在，カーソルがある行を１行消去します．カーソルは，その行の先頭（左端）へ移ります．

⑪ ESC＞BASICステートメント　BASICステートメントを実行します．実行の結果は端末装置側に表示されます．

⑫ ESC. BASICステートメント　BASICステートメントを実行します．実行の結果はどこへも表示しません．

⑬ ESC＜BASICステートメント　BASICステートメントを実行します．実行の結果はホストコンピュータ側に送信され，端末装置には表示されません．

注 意　ESCシーケンスの受信内容はディスプレイ，プリンタのいずれにも表示・出力されません．

(3) リモートBASICプロトコル

前項で説明したスペシャルエスケープシーケンスの中で，⑪⑫⑬を使用すると，ホストコンピュータで作成したBASICプログラムを端末装置側で実行させることができます．

PC-9800シリーズパーソナルコンピュータシステム以外の情報処理装置を一括して，”ホストコンピュータ”とよんできましたが，このホストコンピュータが，PC-9800シリーズパーソナルコンピュータであってもよいわけです．PC-9800シリーズパーソナルコンピュータ間でBASICプログラムの交換をすることもできます．

前項の⑪⑫⑬をあらためて，“リモートBASICプロトコル”とよびます．

⑪ ESC＞BASIC STATEMENT

⑫ ESC. BASIC STATEMENT

⑬ ESC＜BASIC STATEMENT

リモートBASICプロトコルが持つ固有な機能，注意事項について説明します．

(a) 実行結果の表示（固有な機能）

・print文やprint using文など，実行結果を画面に表示させる命令では，表示を端末装置側に行うか，ホストコンピュータ側で行うかを選択することができます．ただし，ホストコンピュータ側へはテキスト画面への表示情報に限って送信されます．送信されたデータの処理はホストコンピュータ側の問題になります．

⑪は，端末装置側（ESC>）

⑫は，出力・表示しない（ESC.）

⑬は，ホストコンピュータ側（ESC<）

・グラフィック命令は，このESCシーケンスの指定に関係なく，必ず端末装置側のディスプレイに実行結果を表示します．

(b)　注意事項

ターミナルモードに切り換わった時，以前の実行モードとの関係に注意してください．

・BASICモードから切り換わった場合

BASICモードの実行中に切り換わった場合は，以前のBASICモードの実行環境と混合しないように注意することが必要です．

たとえば，次のような例に注意してください．

① 以前のBASICモードで使用していたファイルバッファをつぶさないようにするためには，BASICモードで使用していたファイル＃とリモートBASICプロトコルで使用するファイル＃とは異なったものに定義する必要があります．

② リモートBASICプロトコルでメモリ使用条件を定義（clear文）したり状態表示（fre関数）をしたりする場合も，以前のBASICモードとの関数を理解しておくことが必要になります．

(c)　リモートBASICプロトコルの利用例

・端末装置としてのPC-9800シリーズパーソナルコンピュータにグラフを表示させます．

```
ESC>screen 0, 0
ESC>circle (100,50), 100,7
ESC>a=100:b=100:c=200:d=50
ESC>line (100,50) - (a,b), 7
ESC>line (100,50) - (c,d), 7
ESC>paint (120,70), 1, 7
```

・端末装置としてのPC－9800シリーズパーソナルコンピュータのディスクにデータを出力します．

```
ESC>open "2:TEST1" for output as#1
ESC>print #1,100
```

注　意　リモートBASICプロトコルの各文の終端記号（デリミタ）はC_Rを使用します．C_R＋L_Fの場合はL_Fの制御が個別に働きカーソルの移動をともないます．

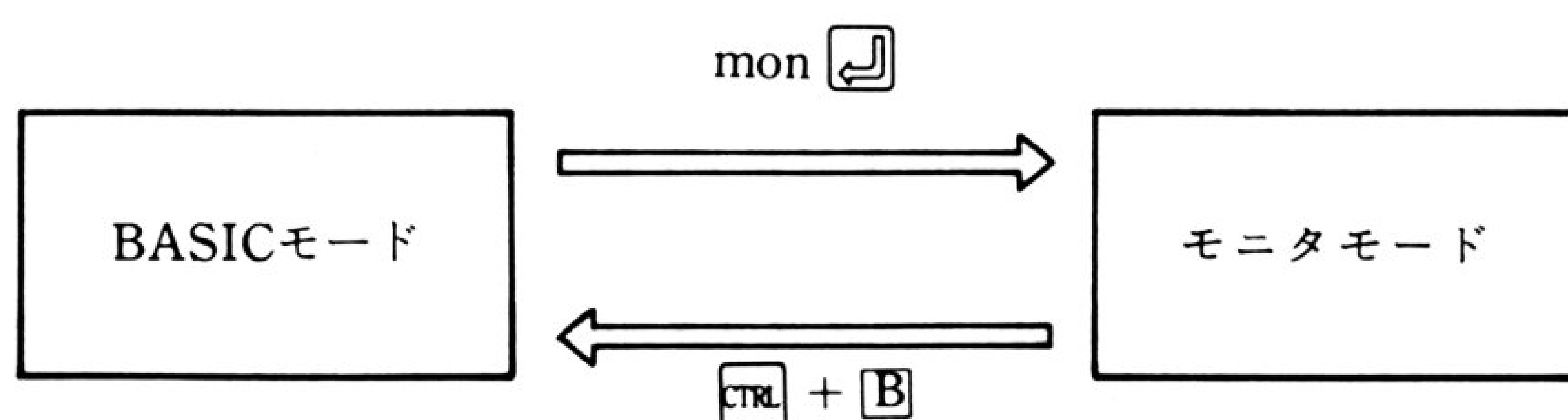
第14章
モニタモード

モニタモードにおいて機械語モニタの使用が可能となります．機械語モニタは，機械語プログラムの作成を手助けするための機能です．

14.1　機械語モニタの動かし方

機械語モニタが機能する環境は，1つの実行モードとして，BASICモードとは別な実行環境をもっています．これを“モニタモード”とよびます．ターミナルモードと同様に，BASICモードから切り換えて使用します．

(1)　実行モードとしての遷移図は，次のようになります．

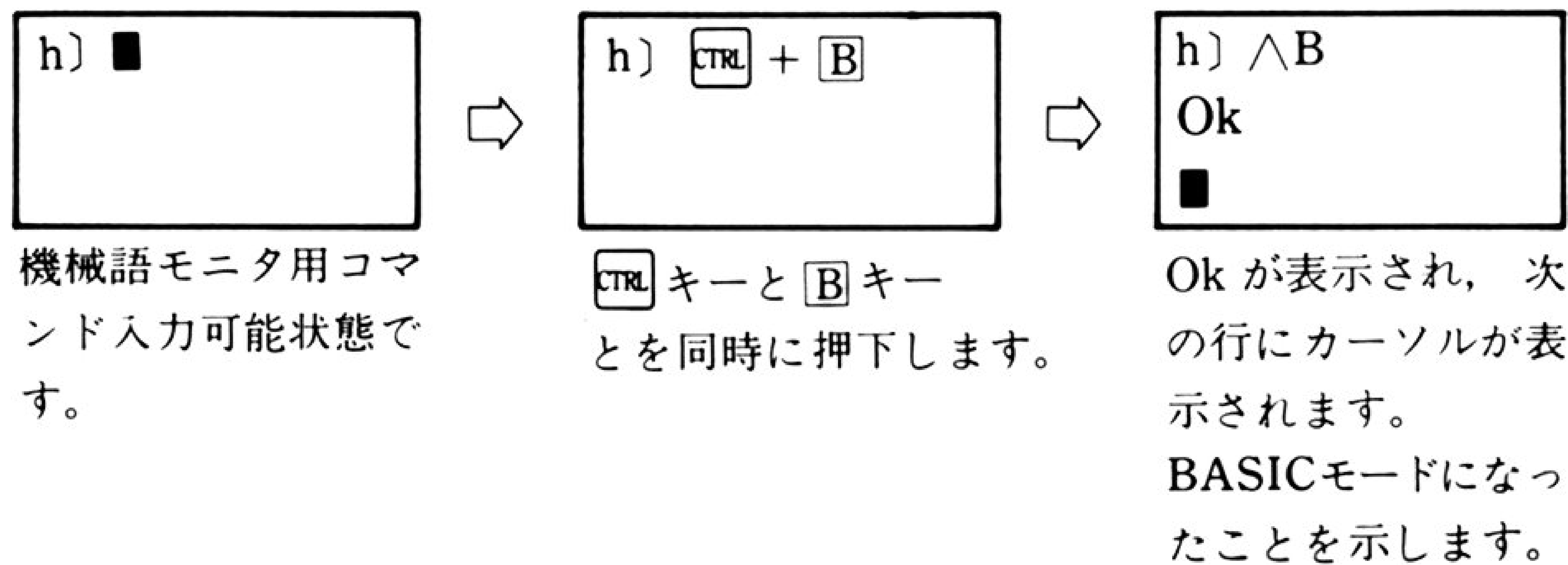


(2)　BASICモードからモニタモードへの切り換え方法



注　意　h］によって，モニタモードになっていることを示します．

(3)　モニタモードからBASICモードへもどるための方法



機械語モニタ用コマンド入力可能状態です。

CTRL キーと B キーとを同時に押下します。

Ok が表示され，次の行にカーソルが表示されます。BASICモードになったことを示します。

14.2 機械語プログラムセグメントのメモリ配置

機械語モニタが処理の対象とするメモリ領域のことを「機械語プログラムセグメント」とよびます．機械語モニタのコマンドは23種ありますが，このコマンドのパラメータで指定するメモリアドレスは，通常機械語プログラムセグメント内のアドレスです．

機械語プログラムセグメントのメモリ上の配置を決める方法を説明します．

(1) BASICモードから，モニタモードに切り換わる直前に機械語プログラムセグメントの配置を決める場合

① clear文でBASICインタプリタの上限を宣言し，次にdef seg文で機械語プログラムセグメントを定義します．

clear, &hm ↵ ……BASICインタプリタが使用するメモリ領域は$(m)_{16}\times16-1$番地まで．

def seg=&hm ↵ ……次の文からのセグメントベースは$(m)_{16}\times16$番地とすることを指定します．

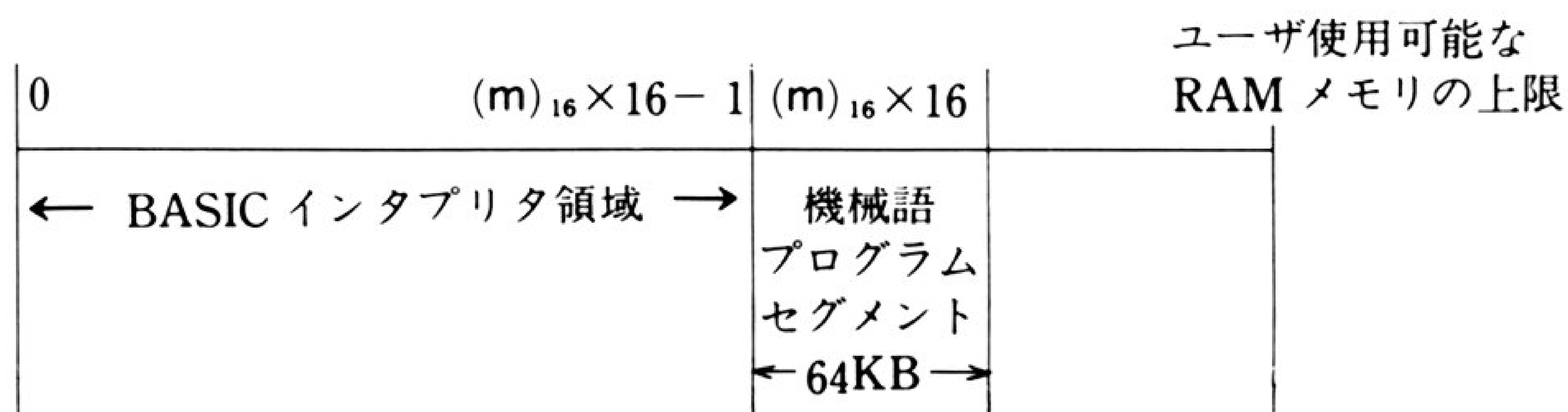
mon ↵ ……機械語モニタモードでは$(m)_{16}\times16$番地をセグメントベースとして使用．

②モニタモード宣言コマンドmonの初期設定値の利用．

clear, &hm ↵

mon ↵

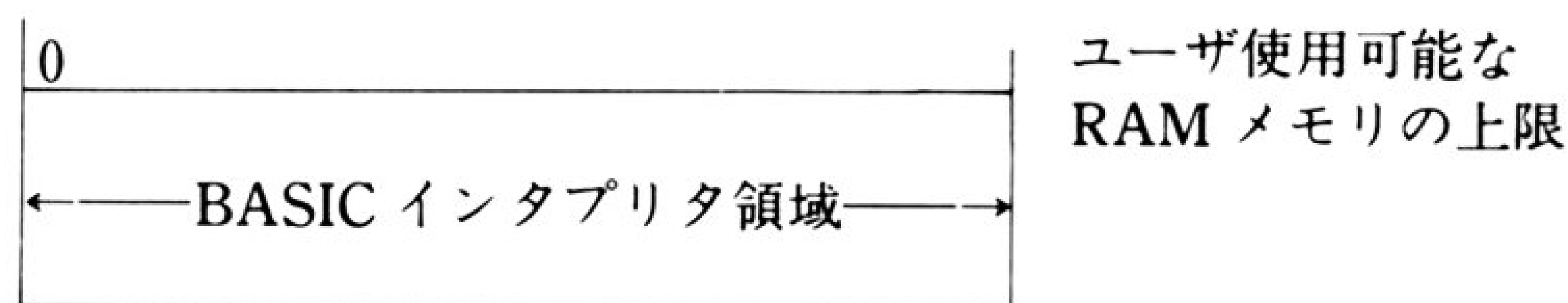
すなわち，def seg＝&hmを実行しないで，monコマンドを使用すると，monコマンドがBASICインタプリタ使用メモリの上限値を，機械語プログラムセグメントベースの値として自動的にとります．



注　意　clear, &hm

def seg = &hm

両コマンドを，ともに使用しないでmonコマンドを実行した場合，BASICインタプリタ領域の上限はユーザ使用可能なRAMメモリの上限となっているので，機械語プログラムセグメントはこの上限からとられます．

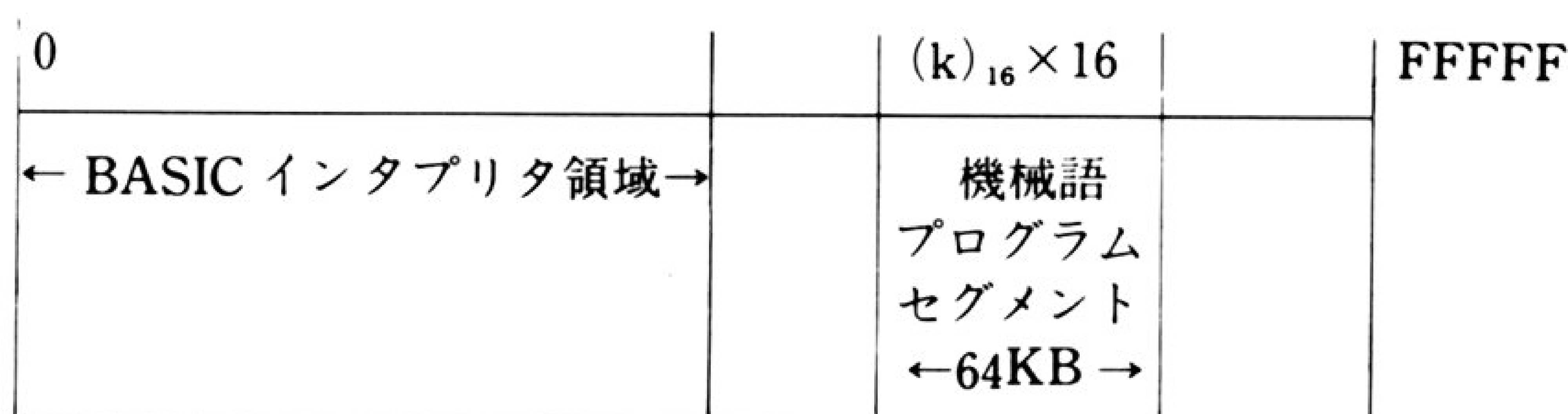


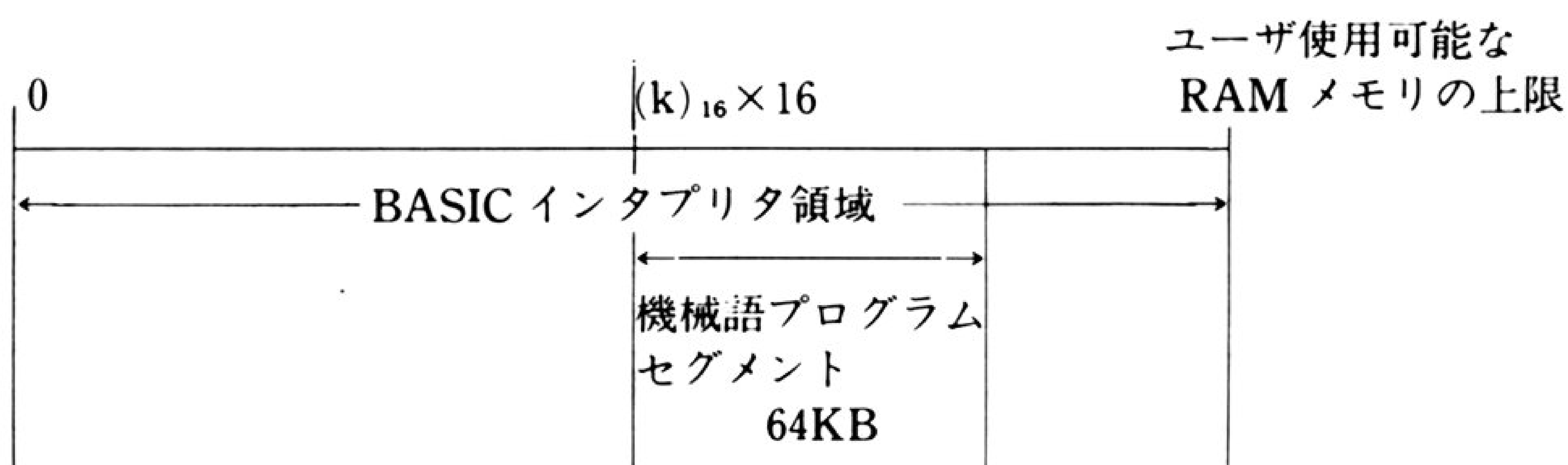
③　機械語セグメントベースアドレスを指定する場合

def seg = &hk ⏎

mon ⏎

とコマンド入力した場合（clear文の第2パラメータでインタプリタ領域の上限を定義しなかったとします），kの値によっては注意が必要です．たとえば，kがBASICシステムの上限値より小さいと，BASICシステム領域と機械語プログラムセグメント領域とが重なり，BASICシステム領域を参照する場合は問題がありませんが，書き込む場合はBASICシステム領域を壊す可能性がありますので注意してください．





(2)　モニタモードの中で，機械語プログラムセグメントの配置を決める場合

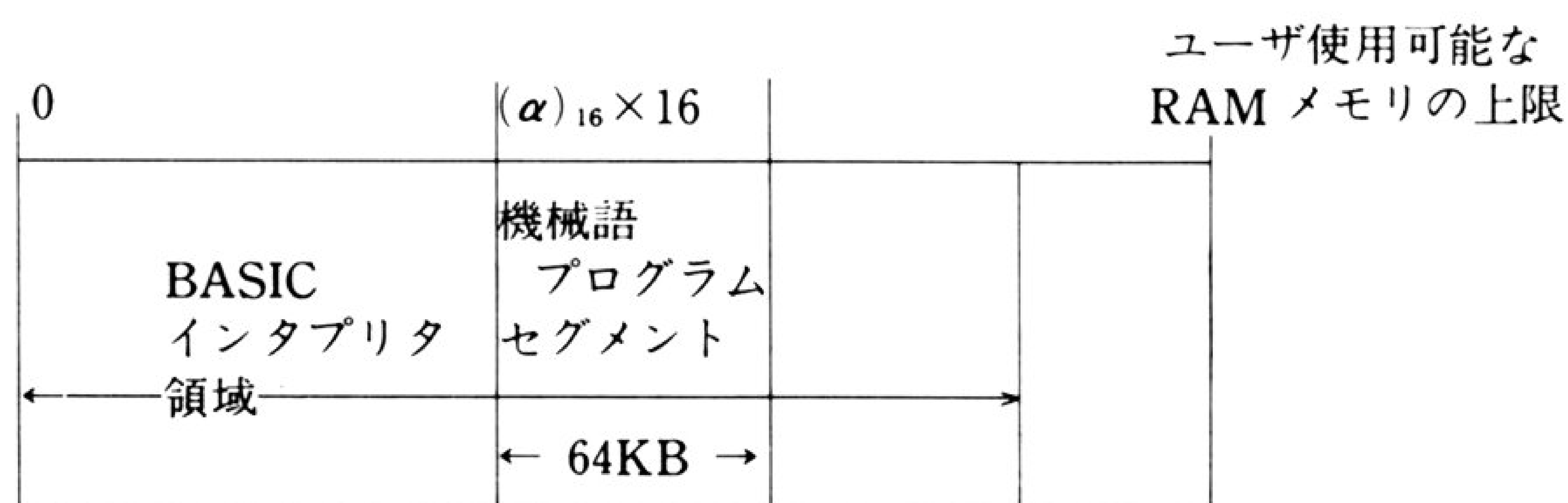
c（チェンジセグメントベース）コマンド

を実行することによって，

モニタモードの中で，セグメントベースを変更することができます．以後のコマンドは指定したセグメント上で機能します．

注 意 1．BASICシステム領域を参照するときは便利ですが，書き込むような場合には，十分確認をとった上で実行してください．

BASICモードに戻ったときの誤動作の原因になります．



αはclearコマンドで指定した機械語プログラムセグメントのベースを指す値です．

14.3 コマンドの種類

機械語モニタ用のコマンドは，次の23種類です．

項番	コマンド	意 味	機 能
1*	A	アセンブル	入力した1行をアセンブルします
2	B	ベース	数値の表現形式を変えます（8進→16進）
3	C	チェンジセグメントベース	機械語プログラムセグメントのベースアドレスを設定します
4	D	ダンプ・メモリ	メモリの内容をディスプレイに表示します
5*	E	エディト・メモリ	スクリーンエディタの機能を用いてメモリの内容を変更します
6	F	フィル・メモリ	メモリの内容を定数で埋めていきます
7	G	ゴー	ユーザープログラムを実行します
8	I	インプット	I/Oポートの値を読み込みます
9*	L	ディス・アセンブル	機械語を逆アセンブルします
10	M	ムーブ・メモリ	ある範囲のメモリの内容を他のアドレスのメモリ領域へ移します
11	O	アウトプット	I/Oポートへデータを出力します
12	P	プリンタ・スイッチ	プリンタへの出力をコントロールします
13	R	リード・テープ	カセットテープからデータをロードします

項番	コマンド	意　味	機　　能
14	S	セット・メモリ	メモリにデータをセットします
15	TM	テスト・メモリ	メモリをテストします
16	V	ベリファイ・テープ	カセットテープの内容と，メモリの内要を比較します
17	W	ライト・テープ	メモリの内容をカセットテープにセーブします
18	X	イグザミン・レジスタ	CPUのレジスタの値を調べ，変更します
19*	HELP または CTRL ＋A	ヘルプ	コマンドとそのパラメータの形式をディスプレイに表示します
20	CTRL ＋B	リターン	BASICモードへ復帰します
21*	CTRL ＋D	ダンプ・ディスク	ディスクの内容をディスプレイに表示します
22*	CTRL ＋R	リード・ディスク	ディスクからデータをロードします
23*	CTRL ＋W	ライト・ディスク	ディスクへデータをセーブします

注意　＊がついている7つのコマンドはDISK BASICモードでメモリスイッチSW 6・2^3ビットがONの場合に限り使用できます（詳細は「2.4 モニタモードにおける拡張機能の選択」を参照してください）

14.4　コマンド使用上の規則

(1)　数値の入力

機械語モニタで扱う数値形式には，16進数モードと8進数モードがあります．この切り換えは，Bコマンドで行います．システム既定値は16進数モードです．

①　16進数モード

アドレス　　：4桁　0～FFFF

メモリの内容：2桁　0～FF

②　8進数モード

アドレス　　：6桁　0～177777

メモリの内容：3桁　0～377

注意1　アドレスの場合は，入力された数値の終りから16ビットが有効になります．メモリの内容の場合は，終りから8ビットが有効になります．

注意2　数値の演算機能があります．

各コマンドでアドレスやメモリの内容を入力する場合，加減の演算機能が使用できます．

(2)　文字列の入力

以下の各コマンドの文字列は先頭から有効桁のみが有効となります．

以降の“，”または“⏎”までは無視されます．

コマンド名	パラメータ	有効桁数
R, V, W	ファイル名	6
B	HまたはQ	1
S	SW1〜SW7	3
X	CPUレジスタ名	2
X	CPUフラグ名	1

(3) カセットテープのスピード

機械語モニタでは，Rコマンド，Vコマンド，Wコマンドで，カセットテープを使用します．カセットテープのスピードは転送速度識別コードで指定します．

1200ボーは「1：」，600ボーは「2：」で示します．省略した場合は1200ボーとなります．

一般形式

R［1：/2：］［〈ファイル名〉］

W［1：/2：］［〈ファイル名〉］，〈セーブ開始アドレス〉，〈セーブ終了アドレス〉

V［1：/2：］［〈ファイル名〉］

(4) ディスクアクセスについての注意

① 複数サーフェスをもつディスクユニットを使用する場合は，サーフェスをまたがってアクセスすることはできません．また，サーフェス番号は必ず指定しなければなりません．

② 1MBフロッピィディスクの0トラックをアクセスすることはできません．

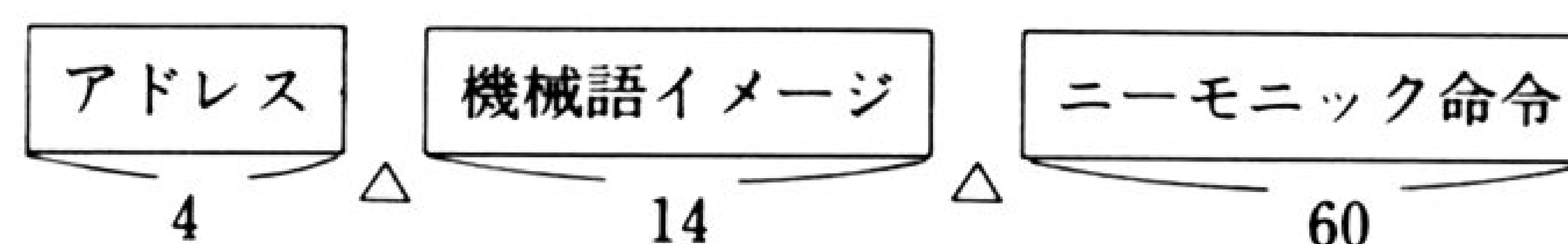
(5) モニタモードのディスプレイ画面形式

① BASICモードからの切り換え直前のWIDTH文による状態が引き継がれます．

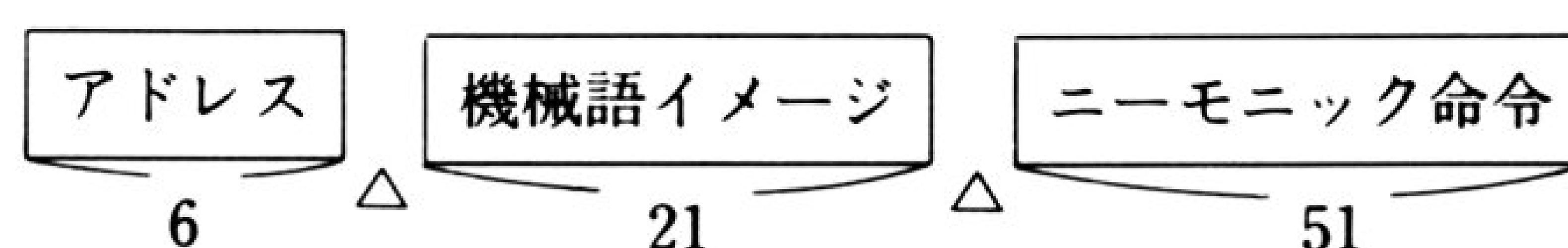
② アセンブル(A)，ディスアセンブル(L)コマンド処理時の画面形式

画面形式は，行当りの文字数（40字/行，80字/行）と数値形式（16進，8進）によって異なります．以下で使用する△は1桁のスペースを示します．

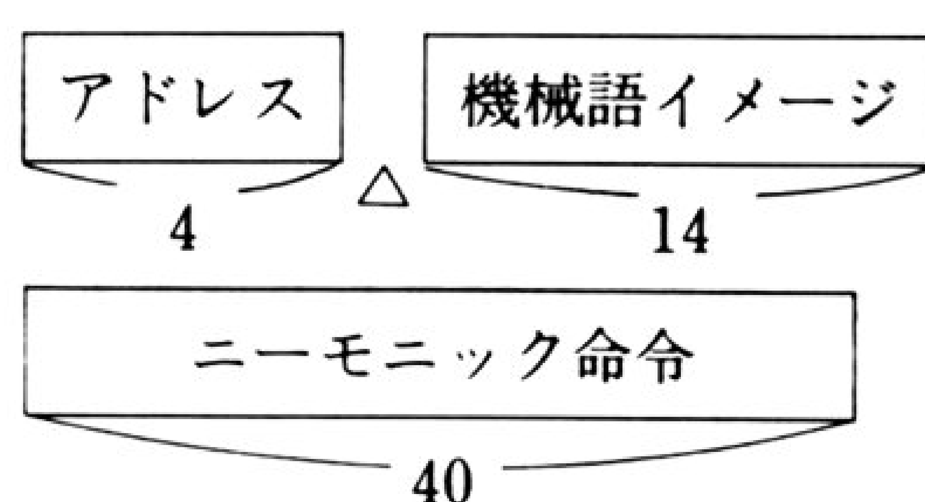
・80字/行，16進数表現



・80字/行，8進数表現

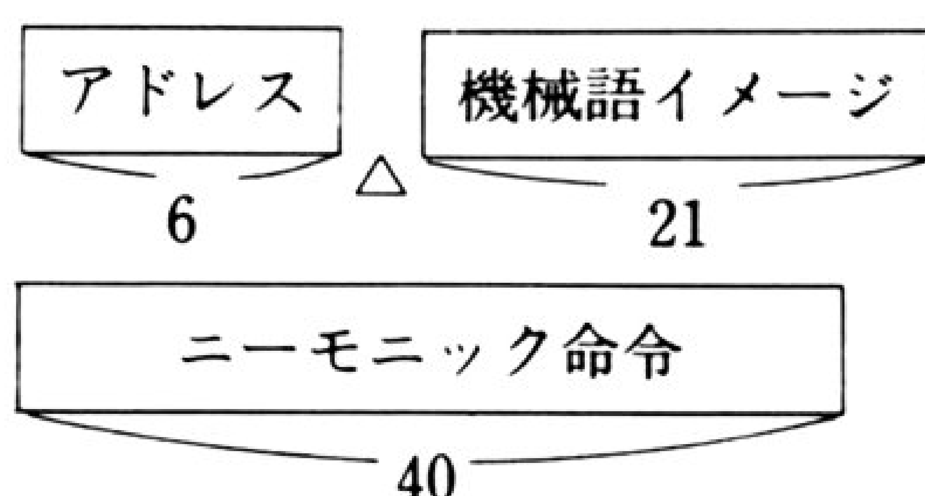


・40字/行，16進数表現



1つの処理単位が2行で表示されます。

・40字/行，8進数表現



1つの処理単位が2行で表示されます。

(6)　メモリアクセスの指定範囲についての注意

アドレスの指定は8進数0～177777，16進数0～FFFFの範囲内で可能です．

開始アドレスから終了アドレスまでの指定範囲については10進数で65535以内でなくてはなりません．

例えば，ダンプの開始アドレスから終了アドレスまでの指定可能な範囲は，$(0)_{16}$～$(FFFE)_{16}$ $(1)_{16}$～$(FFFF)_{16}$です．フィルメモリについても同様な制限があります．

14.5　コマンドの説明

(1)　A（アセンブル）（ROM BASICモードでは使用できません）

入力形式：a ［〈格納開始アドレス〉］

機能：i-8086相当のニーモニックで入力した1行のテキストをアセンブルし，できた機械語をメモリに格納します．そのために，Aコマンドでは，機械語を格納する開始番地を指定します．格納開始番地の指定を省略した場合は，その前に実行したAコマンドの実行終了番地の次の番地となります．また，Aコマンドが実行されていないとゼロ番地から割り当てられます．

規則：

①　入力方法（例で説明します）

```
h]a 9000 ⏎
```

コマンドと格納開始番地を入力します．この例では9000番地です．

```
h]a 9000
9000 △△△△△△△△△△△△△△△△△■
```

割り当てるアドレスを表示し，機械語にアセンブルした結果を表示するスペースをあけ，ニーモニック命令入力を待ちます．

```
h]a 9000
9000△△△△△△△△△△△△△△△△△ mov △ si, di ⏎
```

ニーモニック命令を入力し，リターンキー⏎を押します．

```
h]a 9000
9000△ 8 9 F E△△△△△△△△△△△△ mov △ si, di
9002△△△△△△△△△△△△△△△△△■
```

アセンブルした結果を，アドレスの右側に表示します．次のアドレスを示します．ニーモニック命令入力を待ちます．

このようにして，ニーモニック命令は右側に，対応する機械語の割付けられたアドレスと機械語の内容は左側に表示していきます．

② 入力したニーモニック命令にエラーがあった場合は，次の行の 1 カラムに「？」を表示し，再度同じ格納開始番地を表示し，ニーモニック命令の入力を待ちます．

③ 入力誤りの修正

ニーモニック命令の入力終了は，リターンキー⏎が入力されたところです．その間に，入力誤りがあれば，リターンキー⏎を入力する直前までに修正します．INS，DEL，カーソル移動キーが使えますので，これらによって修正してください．

④ アセンブラから抜け出すには，アドレスが表示された後，リターンキー⏎を入力するか，ストップキーSTOP，またはCTRL＋Cを入力してください．

注　意　64Kバイトを越えるような動作をする命令，アドレス（たとえば，セグメント間CALL/JUMP/RET命令等）をアセンブルすることはできません．

注　意　アセンブラのアドレス割当てについて

機械語プログラムを作成する場合には，def seg文で機械語プログラムセグメントのベースアドレスを宣言しなければなりません．def seg＝&hmのmを16倍した値が機械語プログラムセグメントのベースアドレスとなります．

Aコマンドで機械語プログラムに割当てられるアドレスは，機械語プログラムセグメント内の相対アドレス（オフセット）で示されます．

初期状態では，0番地から割当てられます．

⑵　B（ベース）

入力形式：b $\{{h \atop q}\}$

機能：機械語モニタで，数値を取り扱う形式を設定します．数値をキーボードから入力したり，画面に表示する場合に16進数で行う場合はhを，8進数で行う場合は，qを指定します．現在の表示モードは，促進メッセージの記号で表わされています．

①　bhを入力した場合，またはBASICモードから切り換わった場合は16進数表示です．入力促進メッセージはh］となります．

②　bqを入力した場合は8進数表示です．入力促進メッセージはq］となります．

⑶　C（チェンジセグメントベース）

入力形式：C［〈セグメントベースアドレス〉］

機能：機械語プログラムセグメントのベースアドレスを変更します．〈セグメントベースアドレス〉の値を16倍した値が実際のベースアドレスになります．monコマンド実行直後のベースアドレスはdef seg文で指定した値です．〈セグメントベースアドレス〉を指定しない場合は，現在のセグメントベースアドレスを表示します．

規則：①　モニタコマンドの中で使用するアドレスは，機械語プログラムセグメント内のアドレスです．対応する物理アドレスは，〈セグメントベースアドレス〉の値を16倍した値に，コマンド中に表われるアドレスを加算した値です．

②　Cコマンドでセグメントベースを設定した後，A, D, E, L, Sコマンドにおいて，アドレス指定を省略したときは，セグメント内アドレスは0番地から割当てられます．

⑷　D（ダンプ・メモリ）

入力形式：d［〈表示開始アドレス〉］，［〈表示終了アドレス〉］

機能：メモリの内容をディスプレイに表示するコマンドです．16進モードになっている場合は，その数に対応するアスキー文字も表示されます．さらに，プリンタ・スイッチでプリントモードが指定されていれば，結果がプリンタへも出力されます．（プリンタ・スイッチおよびプリントモードについては，⑿　P（プリンタ・スイッチ）を参照してください）．

規則：①　dコマンドのパラメータ省略時の規則を例で説明します．

(i)　d 9000，9030 [↵]

(ii)　d 9000 [↵]

(iii)　d，9031 [↵]

(iv)　d [↵]

(i)の場合が標準で，9000番地から9030番地までのメモリの内容が表示されます．

(ii)のように〈表示開始アドレス〉だけ指定した場合は，〈表示開始アドレス〉から，16バイト分のメモリの内容が表示されます．

逆に(iii)のように〈表示終了アドレス〉だけ指定した場合は，その前に実行されたDコマンドの最後のアドレスの次のアドレスから〈表示終了アドレス〉までのメモリの内容を表示します．

(iv)のように〈表示開始アドレス〉も〈表示終了アドレス〉も指定しない場合は，このコマンドが入力される前に実行されたDコマンドの最後のアドレスの次のアドレスから16バイト分のメモリの内容が表示されます．

②　Dコマンドを実行中にその実行を中止したい場合は，[STOP]キー，または[CTRL]＋[C]を

入力します．画面にメモリの内容を表示している場合，途中で表示を一時停止させるには，CTRL ＋S を入力します．再び表示を再開させる場合は，STOP キーと CTRL ＋C 以外のキーを入力します．

③ また，Dコマンド終了後にリターンを入力すると(iv)と同じ動作を行います．

④ Dコマンドでダンプできる連続した空間は最大65535バイトです．

(5) E（エディット・メモリ）（ROM BASICモードでは使用できません）

入力形式：e［〈開始アドレス〉］

機能：メモリの内容をディスプレイに表示します．必要ならば，表示されている内容をディスプレイ上で変更します．ディスプレイ上で変更された数値は，そのままメモリ上で変更されます．

規則：① 〈開始アドレス〉を省略した場合は，以前に行ったEコマンドの終了したときに，カーソルがあったアドレスが〈開始アドレス〉となります．

② カーソル移動キー（←，→，↑，↓）とロール・アップキー ROLL UP，ロール・ダウンキー ROLL DOWN を使って変更したいメモリの内容のところへカーソルを移動し，新しい数値を入力してください．メモリの内容も変更されています．

③ エディット・メモリを終了するときは STOP キー，CTRL ＋C キー，または ESC キーを入力します．

④ HELP キーを入力することにより，エディット・メモリ実行時に使用できるキーの説明が表示されます．もとの画面にもどすためには，STOP キー，および CTRL ＋C 以外のキーを入力してください．

注 意 Eコマンドを使用する場合，モニタモードに入る直前のテキスト画面のスクロールエリアが次の範囲になければなりません．

CONSOLE　n，m において，

n＝0

m≧19

(6) F（フィルメモリ）

入力形式：f〈開始アドレス〉，〈終了アドレス〉，〈定数〉

機能：アドレスで指定した範囲のメモリの内容を指定した定数の値で置き換えます．

規則：① 各パラメータは省略できません．

② 〈定数〉は10進数で0から255までの値です（16進数のとき0～FF，8進数のとき0～377）．〈定数〉として入力された数値は，下位8ビットが有効となります．

③ Fコマンドで扱える連続した空間は最大65535バイトです．

(7) G（ゴー）

入力形式：g［〈実行開始アドレス〉］［，〈ブレイクポイントアドレス＃1〉］［，〈ブレイクポイントアドレス＃2〉］

機能：指定したアドレスにジャンプして，そこからプログラムの実行を開始します．このとき，2つまで〈ブレイク・ポイント・アドレス〉を指定できます．〈ブレイク・ポイント・アドレス〉とは，プログラムの実行中，実行アドレスが〈ブレイク・ポイント・アドレス〉となったとき，実行を中止して，コマンドの入力待ち状態にさせるアドレスです．

規則：① 〈実行開始アドレス〉を省略した場合，最後に実行したGコマンドの〈ブレイク・ポイント・アドレス〉が〈実行開始アドレス〉になります．

② 〈ブレイク・ポイント・アドレス〉で実行が中断された場合は，CPUのレジスタの値が保存されます．

③ 〈ブレイク・ポイント・アドレス〉に達したときはそのアドレスが示されます．

16進表示　＊　xxxx：yyyy

8進表示　＊　xxxxxx：yyyyyy

x～x：セグメントベースアドレス

y～y：セグメント内オフセット

(8) I（インプット）

入力形式：i〈ポート・アドレス〉

機能：入力ポートの値（入力データ）を読み画面に表示します．

規則：① 〈ポート・アドレス〉は0～255の値です．

② 〈ポート・アドレス〉を入力後，[⏎]キーを押下します．すると次の行に入力ポートの値が表示され，キー入力待ちになります．

③ [⏎]キーを押下すると処理は終了します．スペースキーを押下すると，次の入力の値を表示します．

④ 〈ポート・アドレス〉として入力された値は終りから8ビットが有効となります．

(9) L（ディスアセンブル）（ROM BASICモードでは使用できません）

入力形式：l［〈逆アセンブル開始アドレス〉］［，〈逆アセンブル終了アドレス〉］

機能：メモリ・イメージを8086系ニーモニックコードに逆アセンブルして表示します．表示形式は「14.4(5) 機械語モニタモードのディスプレイ画面形式」を参照してください．

規則：① Lコマンドのパラメータ省略時の規則は，D（ダンプ）コマンドの場合と同様の規則です．

(i) l9000,9030 [⏎]

(ii) l9000 [⏎]

(iii) l,9030 [⏎]

(iv) l [⏎]

② Lコマンドの実行中に，実行を中断したい場合は，[CTRL]＋[S]を入力してください．実行を再開するためには，[STOP]キー，および[CTRL]＋[C]以外の任意のキーを入力してください．

③ また，Lコマンドで表示されるニーモニックのオペランドは，現在，指定されている数値表現式で表示されます．

④ Lコマンド終了後にリターンキーを入力すると，(iv)の場合と同じ動作になります．

⑤ Lコマンドで逆アセンブルが可能な連続した空間は65535バイトです．

注 意 64Kバイトを越えるような動作をする命令，アドレス（たとえば，セグメント間CALL/JUMP/RET命令等）を逆アセンブルすることはできません．

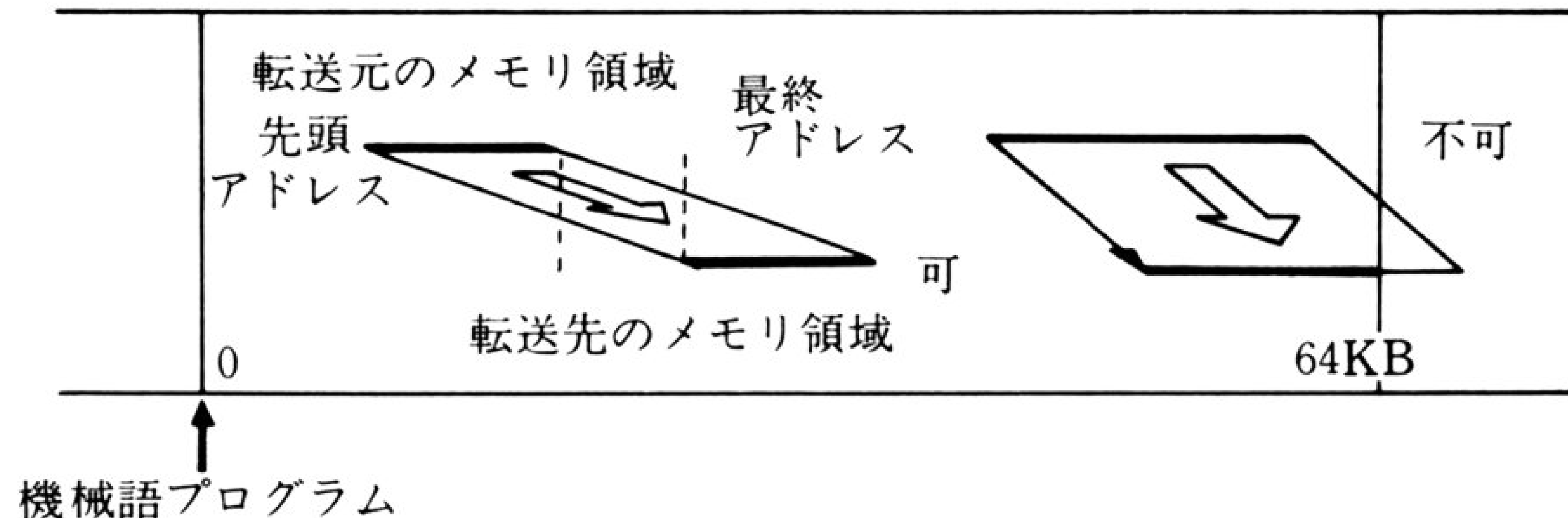
(10) M（ムーブ・メモリ）

入力形式：m〈転送するメモリ領域の先頭アドレス〉，〈転送するメモリ領域の最終アドレス〉，

〈転送先の先頭アドレス〉

機能：メモリ内で，ある位置から別の位置へメモリ領域のブロック転送を行います．

規則：① 各パラメータは省略できません．



② 機械語プログラムセグメントの最大メモリアドレスを越えるような転送はできません．

③ Dコマンドで，転送の結果を確認してください．

(11) O（アウトプット）

入力形式：o〈ポート・アドレス〉,〈データ〉

機能：指定したポート・アドレスに，データを出力します．

規則：〈ポート・アドレス〉,〈データ〉は0～255の値でなくてはいけません．入力した数値は終りから8ビットが有効となります．

(12) P（プリンタ・スイッチ）

入力形式：p

機能：D（ダンプ），L（ディスアセンブル），CTRL＋D（ダンプ・ディスク）コマンドの実行結果をプリンタに出力するかどうかを指定します．モニタモードに切り換わった当初は，「OFF」の状態で，プリンタには出力しません．Pコマンドを入力するたびに，「ON」/「OFF」が切り換わります．

規則：① プリンタスイッチが「ON」になると，コマンド促進のメッセージが変ります．

プリンタスイッチ「OFF」		プリンタスイッチ「ON」
h]	→	h)
q]	→	q)

② D，L，CTRL＋Dコマンド以外の実行結果は，プリンタへ出力されません．

注意 機械語モニタを使用している時も，COPYキーによって画面のハードコピーがとれます．

(13) R（リード・テープ）

入力形式：r［{1 : 2 :}］［〈ファイル名〉］

機能：カセットテープからデータをロードするコマンドです．

ファイル名を指定した場合は，カセットテープのデータのファイル名をチェックし，指定したファイル名が見つかると，そのデータをメモリにロードします．このとき，指定したファイル名以外のファイル名が見つかった場合は，

Skip△：××××

とディスプレイに表示し，指定したファイル名が見つかるまで読み続けます．指定したファイル名が見つかった場合は，

Found：××××

とディスプレイに表示します．

ファイル名を指定しない場合は，最初に見つかったファイルをロードします．

規則：① ファイル名は，6文字以下です．6文字を越えた場合は最初の6文字が有効です．

② コマンド実行中，実行を中断するためには[STOP]，または[CTRL]＋[C]を押下してください．

⒁ S（セット・メモリ）

入力形式：(1) s［〈開始アドレス〉］

(2) s s w［{1〜7}］

機能：形式(1) メモリの内容を表示します．これを確認して内容を変更することができます．

形式(2) メモリスイッチの内容を表示します．これを確認して内容を変更することができます．

規則：形式(1)

① 〈開始アドレス〉を指定した後，リターンキー[↵]を入力すると，次の行にアドレスと内容を表示し，数値の入力待ちになります．

```
h ] S 9 0 0 0 [↵]
9 0 0 0 △ D 3 — ■
```

② メモリの内容を変更する場合は，数値を入力した後スペースキーを入力するとメモリに新しい値をセットします．その後，次のアドレスの内容が表示され，数値の入力待ちになります．また，数値を入力した後リターンキーを入力すると新しい値がセットされコマンド入力待ちになります．

```
h] S9000 [↵]
9000△D3—a3 [↵]
h] ■
```

```
h] S9000 [↵]
9000△D3—a3△D4—■
            └スペースキー入力
```

③ メモリの内容を変更しない場合は，数値を入力せずに，スペースキーを入力してください．次のアドレスに移ります．

```
h] S9000 [↵]
9000△D3—△△△D4—■
         └スペースキー入力
```

④　1つ前に戻したい場合は，CTRL ＋ B を入力してください．

規則：形式(2)

①　スイッチ番号を入力しなかった場合は，メモリスイッチの内容をすべて表示します．

```
h] SSW ⏎
SW1   SW2   SW3   SW4   SW5   SW6   SW7
12    34    00    00    00    00    00
```

②　スイッチ番号（swn）を指定した後，リターンキー ⏎ を入力すると，次の行に，指定したスイッチの内容を表示し，数値の入力待ちになります．

```
h] SSW3 ⏎
00—■
```

```
h] SSW3
00—07  00—■
```

```
h] SSW3
00—07 ⏎
h] ■
```

③　スイッチの内容を変更する場合は，変更する数値を入力した後スペースキーを入力すると，スイッチに新しい値をセットし次のスイッチ番号の内容を表示します．その後数値の入力待ちになります．リターンキー ⏎ を入力すれば，新しい値がセットされ，コマンド入力待ちになります．

④　スイッチの内容を変更しない場合は，数値を入力せずに，スペースキーを入力すれば，次のスイッチの処理に移ります．

```
h] SSW1 ⏎
12—       34—■
```

⑤　スペースキーを有効に使用しながら，1行に全部のスイッチの値をセットすることができます．最後のスイッチの値のセットが終るとリターンして入力待ちになります．

```
h] SSW1 ⏎
12—  34—  00--07 00—  00—  00—  00—
h] ■
```

注　意　このセットスイッチ（SSW）は不揮発性メモリの書き換えを行いますが，BASICシステムへの引き渡しはこのままでは行われません．BASICシステムの再立ち上げによってはじめて引き渡されます．

セットしたメモリスイッチの状態が必要な場合はリセットして，システム立ち上げを行ってください．

⒂　TM（テストメモリ）

入力形式：tm

機能：本体に実装されているRAMメモリをテストします．テスト対象メモリ領域はメモリスイッチSW3・2^0～2^2ビットで指定されているユーザメモリ領域と，システム領域（テキストVRAM，グラフィックVRAM）です．

規則：①　テストメモリのためにモニタモードに入る前に次の準備をしてください．グラフィック画面状態を初期化しテキスト画面状態を80行モードにしておきます．

screen　0，2⏎　　width　80,25⏎

②　メモリ・スイッチによって指定した使用可能なメモリサイズ（Real memory）とシステム立ち上げ時のマイクロ診断によって正常と確認されたメモリサイズ（Active memory）とが一致しない場合にはメモリをテストする前に次のメッセージを表示します．

Real△△△memory△：△×××KB

Active memory△：△×××KB

この場合にはなぜ相違しているかを確認した後，ファンクションキー以外のキーを押すとメモリテストを開始します．

③　テスト終了時，ディスプレイに，異常がなければ，

Test complete！

と表示されます．

テストメモリ終了までの経過時間は次のとおりです．

ユーザメモリエリア384KBの場合：約15分

ユーザメモリエリア512KBの場合：約20分

ユーザメモリエリア640KBの場合：約25分

異常アドレスを検出すると，異常発生のアドレスを16進数5桁で表示します．同時にブザーが鳴ります．

④　テストが終了したならば，リセットスイッチを押してください．

⒃　V（ベリファイ・テープ）

入力形式：v［{1：2：}］［〈ファイル名〉］

機能：カセットテープにセーブされたデータとメモリの内容とが一致しているかどうかを比較するコマンドです．

〈ファイル名〉を指定した場合は指定したファイルとメモリの内容を比較します．〈ファイル名〉を指定しなかった場合は，最初に見つかったファイルとメモリの内容を比較します．比較した結果，カセットテープの内容とメモリの内容が一致した場合はコマンド入力待ち状態となり，一致しなかった場合は“？”を表示してコマンド入力待ちになります．

規則：①　ファイル名は6桁以内です．6桁を越えるときは先頭の6文字を使用します．

②　W（ライトテープ）コマンドでカセットテープに書き込んだ後，メモリの内容を変更しない状態でVコマンドを入力しベリファイしてください．

③　コマンド実行中，実行を中断するためには［STOP］，または［CTRL］＋［C］を使ってください．

⒄W（ライトテープ）

入力形式：w［{ 1 : / 2 : }］［〈ファイル名〉］，〈開始アドレス〉，〈終了アドレス〉

機能：メモリの内容をカセットテープへセーブするコマンドです．

規則：⒀，⒃の規則を参照してください．

例：8000番地から8500番地の内容を“abc”というファイル名でカセットテープにセーブする場合

h］wabc，8000,8500⏎

h］vabc

⒅ X（イグザミン・レジスタ）

入力形式：x { ［cpu レジスタ名］ / ［cpu フラグ名］ }

機能：cpuの全レジスタの内容の表示および変更を行います．

規則：① ［レジスタ名］

AX, BX, CX, DX, SP, BP, SI, DI, CS, DS, SS, ES, IP

② ［cpuフラグ名］

O：Overflow	Z：Zero
D：Direction	A：Auxiliary Carry
I：Interrupt Enable	P：Parity
T：Trap	C：Carry
S：Sign	

③ ［cpuレジスタ名］，［cpuフラグ名］を省略した場合は全レジスタの内容とフラグのステータスを表示します．

④ ［cpuレジスタ名］を指定した場合は，そのレジスタの内容を表示し新しい値の入力待ちとなります．内容を変更する場合は，新しい値を入力した後，リターンキー⏎を押下するとコマンド入力待ちになります．また，スペースキーを押下すると，次のレジスタの処理に移ります．内容を変更しない場合は，新しい値を入力しないでリターンキー（⏎）を押してください．コマンド入力待ちになります．また，スペースキーを入力すると，次のレジスタの処理へ移ります．

⑤ この処理は，不正な値を入力した時または最後のレジスタが表示されるまで繰り返されます．最後のレジスタの処理が終了するとコマンド入力待ちとなります．

⑥ フラグ名を指定した場合は，そのフラグの現在のステータス（「ON」なら“1”，「O-FF」なら“0”）を表示し，新しいステータスの入力待ちとなります．内容を変更する場合には，新しい値を入力してリターンキー（⏎）を押下すると，フラグの内容を変更した後，コマンド入力待ちとなります．内容を変更しない場合には，新しい値を入力しないで，リターンキー（⏎）を押下すると，フラグの内容は変更されずに，コマンド入力待ちとなります．

⑦ レジスタ名の指定は16ビットのペアレジスタのみです．AH, AL…等は使用できません．

⑧ レジスタ名を指定し新しい値として17ビット以上の値が入力されたときは，下位16ビットがとられます．

⑲　X（イグザミン・レジスタ）コマンドでレジスタの値を変更した場合，有効となるのはG（ゴー）コマンドで実行する場合だけです．BASICモードに戻るときは有効ではありません．

(19)　HELPキーまたは[CTRL]＋[Aチ]キー（コマンドシンタックスの表示）（ROM BASICモードでは使用できません）

入力形式：[HELP]，または[CTRL]＋[Aチ]を入力します．

機能：機械語モニタで使用できるコマンドの種類とそのコマンドのパラメータがディスプレイに表示されます．

(20)　[CTRL]＋[Bコ]（BASICモードへの戻り）

入力形式：[CTRL]＋[Bコ]を入力します．

機能：モニタモードからBASICモードに戻ります．

(21)　[CTRL]＋[Dシ]（ダンプディスク）（ROM BASICモードでは使用できません．）

入力形式：[CTRL]＋[Dシ]〈ドライブ＃〉[，〈サーフェス＃〉]，〈トラック＃1〉，
〈セクタ＃1〉[，〈トラック＃2〉，〈セクタ＃2〉]

〈ドライブ＃〉，〈サーフェス＃〉
……ダンプ対象のドライブ番号とサーフェス番号を指定します．複数サーフェスをもつディスクユニットについてはサーフェス番号を必ず指定してください．

〈トラック＃1〉，〈セクタ＃1〉
……表示開始セクタの指定

〈トラック＃2〉，〈セクタ＃2〉
……表示終了セクタの指定

機能：ディスクの内容をディスプレイに表示させます．

規則：①　〈トラック＃2〉〈セクタ＃2〉を指定しない場合は，
〈表示開始セクタ〉＝〈表示終了セクタ〉
となり，開始セクタだけ表示します．

②　[CTRL]＋[D]コマンドを実行中に実行を中止したい場合は，[STOP]キーまたは[CTRL]＋[C]を入力します．画面への表示を一時中断したい場合は[CTRL]＋[S]を入力し，再開するときは[STOP]キーおよび[CTRL]＋[C]以外のキーを入力します．

③　異なるトラックに渡りダンプをおこなうことはできません．

(22)　[CTRL]＋[Rス]（リードディスク）（ROM BASICモードでは使用できません）

入力形式：[CTRL]＋[Rス]〈ドライブ＃〉[，〈サーフェス＃〉]，〈トラック＃〉，
〈セクタ＃〉，〈開始アドレス〉，〈終了アドレス〉

機能：ディスクからメモリへデータをロードします．〈ドライブ＃〉〈サーフェス＃〉〈トラック＃〉〈セクタ＃〉で表示されるディスクの物理アドレスからメモリ上の〈開始アドレス〉～〈終了アドレス〉へデータを読み込みます．

(23)　[CTRL]＋[Wテ]（ライトディスク）（ROM BASICモードでは使用できません）

入力形式：[CTRL]＋[Wテ]〈ドライブ＃〉[，〈サーフェス＃〉]，〈トラック＃〉，
〈セクタ＃〉，〈開始アドレス〉，〈終了アドレス〉

機能：メモリの内容をディスクへセーブします．〈開始アドレス〉から〈終了アドレス〉までの

データをディスクの物理アドレス〈ドライブ#〉〈サーフェス#〉〈トラック#〉〈セクタ#〉を先頭のセクタとして，順次連続した物理アドレスへ書き込みます．

第15章

システムディスクに格納されている ユーティリティプログラム

システムディスクに組み込まれているユーティリティプログラムには次のものがあります．files 文を使用して確認してください．

項番	プログラム名	プログラム呼称	内　　容
1	format.nip	ディスクフォーマッティング	フロッピィディスクのフォーマッティングを行います．
2	backup.n88	バックアップ	同じ種類のフロッピィディスク媒体間でのコピーを行います．全ファイルがコピーされます．
3	setinf.n88	IDセクタ書き換え	オートスタートのために，使用するファイル数，BASICテキストをIDセクタへ書き込みます．
4	xfiles.n88	ファイル転送	フロッピィディスクおよび固定ディスク間でのコピーを行います．
5	sysgen.nip	DISK CODE コピー	DISK CODE, IPLおよび日本語辞書ファイルのコピーを行います．
6	setup.n88	日本語入力環境条件の設定	システムディスクの日本語入力環境を規定します．
7	format.hd	ボリューム初期化	固定ディスクの物理フォーマッティングと論理フォーマッティングを行います．
8	recov.hd	障害ボリューム・ファイル復旧	固定ディスクのエラー発生クラスタを正常な未使用クラスタで代替することを行います．
9	dir.hd	ファイルディレクトリ表示	固定ディスクに登録されているファイルのディレクトリを表示したり，削除したりすることを行います．

項番	プログラム名	プログラム呼称	内　　容
10	backup.hd	固定ディスクファイル退避/復旧処理	固定ディスクとフロッピィディスク間でファイルの退避/復旧を行います（1ファイルにつき複数フロッピィディスク使用可）
11	mkfont.n88	利用者定義文字格納ファイル作成・更新	利用者定義文字格納ファイル（ファイル名は“usfontn88”）の作成及び更新を行います.
12	switch.n88	メモリスイッチの設定	メモリスイッチの設定を行います.
13	dicmen.n88	辞書ファイルの保守	辞書ファイルへの単語登録・更新等を行います
14	DDconv.n88	ファイル変換	5”1D/2D媒体上のファイルを2DD/2HD媒体上のファイルに変換します.

この節では，これらのユーティリティプログラムの使い方を説明します.

なお，説明は次に示す環境を前提におこないます.

(1) 本体内蔵のFD装置♯1がドライブ番号「1：」，本体内蔵のFD装置♯2がドライブ番号「2：」であることを前提とします.

(2) システムディスクは本体内蔵のFD装置（ドライブ番号「1：」）から起動され，各種ユーティリティプログラムもシステムディスク内に格納されていることを前提とします.

15.1 フロッピィディスクのフォーマッティング

ディスクを初期化することは，「フォーマッティングする」と言います．フォーマッティングには，「イニシャライズ」と「システムフォーマッティング」があります.

(1) イニシャライズ

イニシャライズを行うと，ディスクユニットで読み書きができるようになります．また，BASICからも，**dski$, dsko$**で入出力を行うことができます．しかし，他のコマンド，ステートメントを使用することはできません．このイニシャライズでは，まだFAT，ディレクトリ，IDの初期化が行われていないからです.

(2) システムフォーマッティング

システムフォーマッティングを行うと，FAT，ディレクトリ，IDの初期化が行われます．これで，そのディスクに対してBASICのコマンド，ステートメントを実行できるようになります.

“イニシャライズ”を「第1レベルのフォーマッティング」または「物理的フォーマッティング」，“システムフォーマッティング”を「第2レベルのフォーマッティング」と呼ぶことがあります.

フロッピィディスクのフォーマッティングは，システムディスクに入っている“format.nip”というプログラムを使います．その手順は次のようになっています．

(1) フォーマッティングプログラム“format.nip”をロードします．

load　“format.nip” [↵]

(2) プログラムを実行させます．

[f･5] または **run** [↵]

(3) 次のメッセージが表示されます．

新しいディスクをドライブにセットして下さい

フォーマットするドライブは？

(4) 適当なドライブに新しいディスクをセットします．(他のドライブには，ディスクを入れないでください．システムディスクはプログラムのロードが終った事を確認したら，はずしておきます.)

(5) 新しいディスクがセットできたら，セットしたドライブのドライブ番号を入力します．ドライブ番号が2の場合は次のようになります．

2 [↵]

(6) すると，次のメッセージが表示されます．

物理フォーマッティングを行いますか（y/n）？

新しいフロッピィディスクでイニシャライズ（物理的フォーマッティング）を行う必要がある場合は，

y [↵]

一度使用したもので，その必要がない場合は，

n [↵]

を入力します．

“n”を入力するとイニシャライズは行いません．

(7) “y”を入力すると，使用装置が1MB/640KB両用タイプのFD装置の場合，次のメッセージが表示されます．(これ以外の装置の場合，このメッセージは表示されません．すぐにイニシャライズが始まります)．

ディスクタイプ（1.1MB　2.640KB）？

これはフォーマットの対象媒体が2HDか2DDかを聞いているもので，2HD媒体であれば，

1 [↵]

と入力します．次のメッセージを表示しイニシャライズが始まります．

ドライブ2を物理フォーマット中です

(8) 次にシステムディスクを作るかどうかたずねてきます．

システムディスクを作成しますか（y/n）？

システムディスクを作りたい場合は，

y [↵]

その必要がない場合は，

n [↵]

を入力します．

(9) システムディスクを作る，作らないにかかわらず，

処理中です

と表示して，システムフォーマッティングを行います．

⑽ システムディスクを作らない場合は，

終了しました

と表示して実行を終了します．

システムディスクを作る場合は，次のメッセージが表示されます．

システムディスクのドライブ番号は？

⑾ システムディスクをセットしたら，そのドライブのドライブ番号を入力します．ドライブ番号が1の場合は次のようになります．

1 [↵]

⑿ システムディスクをセットしたドライブを指定すると，

確認しましたか（y/n）？

と求めてきます．もう一度確認して正しければ，

y [↵]

もしまちがっていれば，

n [↵]

を入力します．

"n" を入力すると，**終了しました**と表示し，処理を終了します．**"y"** を入力すると，

システムをコピーしています

と表示して，DISK codeのコピーを開始します．

なお，⑸で指定した新しいディスクのドライブ番号と，⑼で指定した，システムディスクのドライブ番号が等しい場合には，

受け側ディスクをドライブXにセットして下さい

および

送り側ディスクをドライブXにセットして下さい

が表示されます（Xはドライブ番号を意味します）．それぞれの指示に従って新しいディスクかシステムディスクをドライブXにセットし，[↵] キーを押して下さい．

⒀ 次に日本語入力用辞書ファイルのコピーをおこなうか否かを問い合せてきます．

辞書ファイルをコピーしますか（y/n）？

日本語入力用辞書ファイル（"BUNSET.SU"）のコピーが必要な場合y [↵] を，必要がない場合n [↵] を応答します．

⒁ コピー処理が終了すると，次に，各種フロッピィディスク装置のデバイス番号の割り振り方をたずねてきます．

システムディスクのドライブアロケーションタイプ：1

新しいディスクのドライブアロケーションタイプは？

システムディスクと同じ場合には，[↵] キーだけを押します．

システムディスクとは異なるデバイス番号の割り振り方を新しいディスクにセットしたい場合は，

X [↵]

を入力して下さい．

Xの値により，新しいディスクは，以下の順番で各装置のドライブ番号を割り振るシステムディスクとなります．

Xの値	意　　　味
1	640KBフロッピィディスク→1MBフロッピィディスク
2	1MBフロッピィディスク→640KBフロッピィディスク

⒂ **終了しました**が表示され，処理が終了します．

フォーマッティングは終了です．

新しいフロッピィディスクをシステムディスクとしてフォーマッティングした場合，そのフロッピィディスクでBASICをスタートさせることができます．

注　意　システムディスクを作成する場合，システムディスクとフォーマッティングするフロッピィディスクが同じ種類(同じ容量)のフロッピィディスクでなければなりません．

(NOTE) **フォーマッティングを行うと，そのディスクに書き込まれていたファイルはすべて消去されます．ですから，うっかり重要なディスクをフォーマッティングしたりすることのないように注意して下さい．**

注　意　属性の宣言で書き込み禁止を宣言していても（「11.3.6ファイルの属性とset文」を参照），フォーマッティングを行うと，ファイルはなくなります．3.5インチおよび5インチフロッピィディスクで内容を消したくない場合は，「ライト・プロテクト」という方法があります．各フロッピィディスクユニットについているユーザーズマニュアルをお読みください．8インチ標準フロッピィディスクにはこの機能はありません．

15.2 フロッピィディスクのバックアップ

ディスクにセーブされたファイルは壊れることがあります．例えば，まちがって必要なファイルをkill文で消してしまったり，DSKO$命令でファイルの内容を壊す可能性もあります．

そこで，重要な情報が記録されているディスクは，定期的に複写を作っておいて，もしファイルが壊れても，その複写ファイルを用いて，その被害を最小限にすることが必要です．このような複写ファイルを作ることを「バックアップファイルを作る」と言います．順を追って「バックアップファイルの作り方」を説明します．

○バックアップファイルの作り方

システムディスクには，**“backup.n88”**というバックアップファイル作成プログラムが入っています．このプログラムでは，同じ種類のフロッピィディスク間でのバックアップをおこないます．異なる種類の媒体間でのバックアップはできません．

(1) バックアップファイル作成プログラム **“backup.n88”** をロードします．

load　“backup.n88” [↵]

(2) このプログラムを実行します．

[f･5] またはrun [↵]

(3) すると画面に次のメッセージが出力されます．

ディスクのバックアップを行います

送り側ディスクをドライブにセットして下さい

ドライブ番号は？

バックアップファイルを作りたいディスク（マスターファイル）を適当なドライブにセットし，そのドライブ番号を入力します．ドライブ番号が1の場合は，

1 ⏎

と入力します．

(4) 次に新しいディスク（バックアップファイル）をドライブにセットするように指示してきます．

受け側ディスクをドライブにセットして下さい

ドライブ番号は？

そこで，新しいフロッピィディスク（または，すべて消してもかまわないフロッピィディスク）をセットし，そのドライブ番号を入力します．ドライブ番号が2の場合ですと，

2 ⏎

と入力します．このときマスターファイルとバックアップファイルを作るフロッピィディスクの種類が合わない場合は，

バックアップできません

と表示されます．このときはマスターファイルとバックアップファイルのフロッピィディスクの種類が異っているか．ドライブ番号の入力がまちがっています．

(5) さらに次のメッセージが表示されます．

ディスクのバックアップを行います

物理フォーマッティングを行いますか（y/n）？

すでにイニシャライズ（物理的フォーマッティング）が行われているフロッピィディスクを使用する場合は

n ⏎

を入力します．すると(6)へ進みます．

新しいフロッピィディスクでイニシャライズ（物理的フォーマッティング）が必要な場合は，

y ⏎

を入力します．

(6) 続いてバックアップ処理が始まります．マスターファイルの入っているドライブとバックアップファイルを作るドライブのアクセス表示用LEDが順に点灯すると同時に，画面にコピーしているトラック番号が表示されます．

XXトラックコピー中です

XXはコピー中のトラックの番号を意味します．

なお，(3)で指定したマスターファイルのドライブ番号と，(4)で指定したバックアップファイルのドライブ番号が等しい場合には，コピーしているトラック番号を表示している途中で，

送り側ディスクをドライブXにセットして下さい

および

受け側ディスクをドライブXにセットして下さい

が表示されます（Xはドライブ番号を意味します）．それぞれの指示に従ってマスターファイルかバックアップファイルをドライブ番号Xにセットし，[↵]キーを押して下さい．

(7) 最後に**終了しました**が表示され，処理は終了します．

15.3 IDセクタの書き換え

IDセクタを書き換えることにより，BASICをスタートさせると同時にファイル数がセットされ，任意のプログラムを自動的に実行することができます．このために，「IDセクタ書き込みプログラム（setinf.n88）」が用意されています．

以下の操作例はフロッピィディスクに対するIDセクタ書き換えの例です．

(1) まず，**"setinf.n88"** をロードします．

load "setinf.n88" [↵]

(2) プログラムを実行させます．

[f･5] あるいはrun [↵]

(3) すると画面に次のメッセージが表示されます．

オートスタート情報をセットします

システムディスクがセットされているドライブ番号は？

IDセクタを書き換えたい（オートスタート情報を設定したい）システムディスクを適当なドライブにセットしそのドライブ番号を入力します．たとえば，

1 [↵] と入力します．すると，

確認しましたか（y/n）？

と表示されますので，

y [↵]

と入力します．

(4) How many files (0−15) ?

とたずねてくるので，同時にオープンしたいファイル数を入力します．例えば，3ファイルを同時にオープンさせたい場合ですと

3 [↵]

と入力します．

このとき，−1を入力するとIDのファイル数には255（16進でFF）がセットされます．以降スタートするたびにファイル数を尋ねるようになります．この場合にはオートスタートにはなりませんので以下の操作は意味がありません．

(5) 次に

テキストを入力して下さい

と表示されますから，BASICのスタートと同時に実行したいBASICのテキスト（コマンドやステートメント）を入力します．例えば，"test.n88" というプログラムをスタートさせたい場合

ですと，

run"test.n88" [↵]

と入力します．

もし，ここで入力したBASICのテキストが長すぎた場合は，

テキストは253文字までです

と表示してブザーを鳴らし，もう一度，BASICのテキストの入力を行うように指示してきます．

(6) 次に今入力したファイル数とBASICのテキストを表示します．今の場合ですと，

ファイルの同時オープン数は[3]です

テキストは：[run "test.n88"]

と表示し，続けて，

システムディスクをドライブ1にセットして下さい

準備ができたらリターンキーを押して下さい

と表示してきますから，[↵]キーを入力します．

(7) すると，

終了しました

と表示され，IDセクタにBASICテキストを書き込み，実行を終了します．

これで，このシステムディスクを使用してBASICをスタートさせると，スタートと同時に自動的に，プログラム"test.n88"を実行します．（もちろん，このシステムディスクにはプログラム"test.n88"があるとします．）

なお，固定ディスクのID部に対する情報設定もsetinf.n88を使用します．固定ディスクからオートスタートをおこなう場合，ID部に対する情報の設定以外にDISK code，IPLおよび日本語辞書ファイルを固定ディスクへ設定しなければなりません（この方法に関してはsysgen.nipあるいはformat.hdを参照してください）．

注 意 (4)で，－1を入力した場合は，(6)で**立ち上げ時にファイルの同時オープン数を指定します**と表示します．また，この場合はオートスタートではなく，一般のシステム立ち上げ処理を行いますので注意してください．

15.4 フロッピィディスク上のファイル転送

「15.2 フロッピィディスクのバックアップ」で説明した"backup.n88"を使用する場合は，バックアップファイルは同じ種類のディスクにしか作成できませんでした．ここでは，異なったディスク間のファイルのバックアップを行うプログラムについて説明します．

このプログラムは，フロッピィディスクおよび固定ディスク間でファイルをコピーします．

なお，媒体容量の差異によって，移行時に容量制限が越える場合があります．この場合には"disk full"のエラーメッセージが表示されます．

注 意 xfiles.n88を使用して固定ディスク内のファイルをフロッピィディスクに転送する場合，一枚のフロッピィディスクの容量を越えたファイル転送はできません．一枚のフロッピィディスクの容量を越えたファイルの転送は"backup.hd"を使用してくださ

い．

移行の対象範囲についてボリューム単位またはファイル単位の選択が可能です．

以下，順を追ってこのプログラムの使い方を説明します．

(1)　システムディスクから“xfiles.n88”をロードします．

load“xfiles.n88”[⏎]

(2)　プログラムを実行させます．

[f･5] あるいは run [⏎]

(3)　すると画面に次のメッセージが表示されます．

ファイル転送を行います

送り側のドライブ番号は？

送り元のドライブ番号を問い合わせてきます．

(4)　送り元（マスタになるところ）のドライブ番号を入力します．たとえば，

1 [⏎] と入力します．

(5)　送り先（コピーをとるところ）のドライブ番号を問い合わせてきます．

受け側のドライブ番号は？

(6)　送り先のドライブ番号を入力します．たとえば，

3 [⏎] と入力します．

(7)　次に　ファイル単位(f)の転送かボリューム全体(v)の転送かを問い合わせてきます．

ファイル単位ですか．ボリューム単位ですか（f/v）？

(8)　ファイル単位のコピーの場合f [⏎] あるいはボリューム単位のコピーの場合v [⏎] と入力します．

(9)　(8)でfと応答した場合，画面にファイル名の一覧と次のメッセージが表示されます．

転送するファイルを選択して下さい．矢印キーでカーサ移動後リターンキーを押して下さい．ROLL UPキーを押すと次の画面になります．

(10)　メッセージに従ってファイルを選択して下さい．

(11)　ファイルが選択されるとコピーが始まります．この時そのファイル名はリバースで表示され，画面下には**転送中です**と表示されます．

(12)　別のファイルのコピーを続けておこなうかどうかを問い合わせてきます．

別のファイルを転送しますか（y/n）？

別のファイルがあればyを入力します．

すると(9)からの繰り返しになります．

ファイルのコピーを終了する場合はnを入力してください．

(13)　転送が終了すると

終了しました

と表示されます．

(14)　(8)でvと応答した場合にはボリューム単位でコピーを行っていいかの確認を要求してきます．

ボリューム単位でよろしいですか（y/n）？

この時ドライブ番号の確認も行って下さい．

確認したらyと入力して下さい．するとコピーが始まり，コピー中のファイル名はリバースで表示され，画面下には**転送中です**と表示されます．

ボリューム内のすべてのファイルがコピーされると**終了しました**と表示され，処理終了となります．

15.5　DISK code, IPLおよび日本語辞書ファイルのコピー

このプログラムはフロッピィディスクあるいは固定ディスクにDISK code, IPLおよび日本語辞書ファイルをコピーします．固定ディスクからBASICを立ち上げる場合，このプログラムを使用して，BASICのDISK code部とIPLを固定ディスクに格納しておかなければなりません（日本語入力をおこなう場合，日本語辞書ファイルも必要です）．また，システムディスクのDISK code部が何らかの理由で損傷を受けた場合，このプログラムを使用してDISK code部を回復することができます．

(1)　DISK codeのコピープログラム“sysgen.nip”をロードします．

　　load　“sysgen.nip”　[↵]

(2)　プログラムを実行させます．

　　[f･5] または run [↵]

(3)　次のメッセージが画面に表示されます．

　　システムをコピーします

　　送り側のドライブ番号は？

(4)　システムディスクがセットされているドライブのドライブ番号を入力します．

　　1 [↵]

　　と入力します．

(5)　すると次のメッセージが表示されます．

　　受け側のドライブ番号は？

(6)　DISK code, IPL，日本語辞書ファイルの書き込みを行いたいフロッピィディスクがセットされているドライブのドライブ番号または固定ディスクのドライブ番号を入力します．ただし，このときフロッピィディスクまたは固定ディスクは，あらかじめフォーマッティングされていなければなりません．また，フロッピィディスク間で処理をおこなう場合には，両方のフロッピィディスクは同じ種類のものでなければなりません．

　固定ディスク＃1にDISK CODE, IPLおよび日本語辞書ファイルを設定する場合，たとえば，

　　3 [↵]

　と入力します．

(7)　次に

　　準備ができたらリターンキーを押して下さい

　と表示されますから，

　　[↵] キーを入力します．

　もし，まちがっていれば，[STOP] キーを入力してプログラムを止め，もう一度，(2)からスタートします．

(8)　**コピー中です**

と表示され，DISK code，IPLのコピーを開始します．

(9)　次に日本語入力用辞書ファイルのコピーをおこなうか否かを問い合せてきます．

辞書ファイルをコピーしますか（y/n）？

日本語入力用辞書ファイル（“BUNSET.SU”）のコピーが必要な場合y ⏎を，必要がない場合n ⏎を応答します．

(10)　最後に，システムディスクに付与される各種フロッピィディスク装置のデバイス番号の割り振り方をたずねてきます．

システムディスクのドライブアロケーションタイプ：X
新しいディスクのドライブアロケーションタイプは？

もとのシステムディスクと同じ場合には ⏎ キーだけを押します．

もとのディスクとは異なるドライブ番号 の割り振り方を書込みを行うディスクにセットしたい場合には，

X ⏎

を入力して下さい．

値Xとデバイス番号の割り振り方の対応については、「15.1　フロッピィディスクのフォーマッティング」の手順(14)を参照してください．

(11)　**終了しました**が表示され，処理が終了します．

注　意　**固定ディスクへDISK code，IPLおよび日本語辞書ファイルをコピーする場合，BASICの領域が固定ディスク内の先頭に位置していないと，他のOS（たとえば，MS-DOS）の領域を破壊してしまいます．したがって，固定ディスクからBASICを立ち上げるためにDISK codeとIPLの設定をおこなう場合，他のOSのためにフォーマッティングをおこなう前に，必ずBASICのためにフォーマッティングをおこなってください．**

15.6　日本語入力環境条件の設定

このプログラムは次の属性をシステムディスクに与えます．

(1)　そのシステムディスクで取り扱うことのできる日本語入力の方法

(2)　日本語入力時に参照する辞書ファイルのドライブ番号

N_{88}-日本語BASIC(86)(Ver 4.0)で取り扱うことのできる日本語入力の方法は次の3種類です．

①　連文節変換入力方式
②　単文節変換入力方式
③　JISコード入力方式

日本語入力の方法はシステムディスクにより一意に決まります．

例えば，連文節変換入力用のシステムディスクを使用する場合，連文節変換入力方式による日本語入力しか行えません．

同様に単文節変換入力用のシステムディスクを使用する場合，単文節変換入力方式による日本語入力しか行えません．JISコード入力用システムディスクにおいても同様です．

“setup.n88”ユーティリティはシステムディスクに記録されている日本語入力方法の規定に関する情報を変更します．

システムディスクは連文節変換入力用に特殊化された状態で出荷されます．このシステムディスクを単文節変換入力用システムディスクあるいはJISコード入力用システムディスクに変更したい場合，“setup.n88”ユーティリティを使用します．

もちろん単文節変換入力用のシステムディスクを連文節変換入力用に戻したり，JISコード入力用システムディスクを単文節変換入力用システムディスクに変更したりすることも可能です．

また，“setup.n88”ユーティリティで辞書ファイルの存在するドライブを固定的に宣言することができます．

“読み”を“日本語の並び”に変換する際，システムは辞書ファイルを参照しますが，この時次の規則に従って辞書ファイルを探します．

① 標準のシステムディスクを使用する場合（出荷時の状態で使用する場合），システム起動時にシステムディスクがセットされていたドライブに対し辞書ファイルを探しにいきます．

② 辞書ファイルのドライブ番号を固定的に宣言しておくこともできます（この指定はJISコード入力方式の場合意味を持ちません）．

“setup.n88”ユーティリティで辞書ファイルが存在するドライブ番号をシステムディスクに与えます．このシステムディスクでシステムを起動しますと，指定されたドライブに対して固定的に辞書ファイルを探しにいきます．

(1) **“setup.n88”**をロードします．

load “setup.n88” [↵]

(2) プログラムを実行させます．

[f·5] **またはrun** [↵]

(3) 次のメッセージが表示されます．

システムディスクがセットされているドライブは？

属性を変更したいシステムディスクを適当なドライブにセットし，そのドライブ番号を入力します．例えば，

1 [↵]

(4) 次のメッセージが出力されます．

現在の日本語入力属性は次の通りです

日本語入力方式：連文節変換入力方式注(1)　**辞書ファイルのドライブ：SYSTEM**注(2)

属性を変更しますか（y/n）？

注(1)：システムディスクの日本語入力の方法に応じて次のいずれかが表示されます．

連文節変換入力方式

単文節変換入力方式

JISコード入力方式

注(2)：日本語変換の際参照する辞書ファイルのドライブ番号です.

SYSTEM　：システム起動時にシステムディスクがセットされていたドライブ(標準値)

1　　　　：ドライブ1

2　　　　：ドライブ2

⋮

なお，日本語入力方式がJISコード入力方式の場合，ここには何も表示されません.

nを応答すると処理終了となります.

(5)　**y**を応答すると次のメッセージが表示されます.

日本語入力方式（1．連文節変換入力　2．単文節変換入力　3．JISコード入力）？

例えば，日本語入力方式を単文節変換入力方式に変更したい場合，次のように入力します.

2 [↵]

なお，[↵]キーのみを応答した場合，日本語入力方式は変更されません．3を選択した場合，(7)の確認メッセージの表示に進みます.

(6)　1あるいは2を選択した場合，次のメッセージが表示されます.

辞書ファイルのドライブ番号は？

システムは日本語変換の際，ここで指定したドライブに対して固定的に辞書ファイルを探しにいきます．例えば，次のように入力します.

1 [↵]

この場合，辞書ファイルはドライブ1にセットされている媒体上にあるものとみなされます.

なお，システム起動時にシステムディスクがセットされていたドライブを辞書ファイルドライブとしたい場合，[↵]キーのみを応答して下さい.

(7)　次の確認メッセージが表示されます.

日本語入力方式：単文節変換入力方式注(1)　**辞書ファイルのドライブ：1**注(2)

変更してよろしいですか（y/n）？

注(1)：(5)で選択した入力方式が表示されます.

注(2)：(6)で選択した辞書ファイルのドライブです.

なお，JISコード入力方式の場合，表示されません.

yを応答すると，システムディスクの属性を変更します.

属性を変更しましたというメッセージが表示され処理終了です.

nを応答すると，(4)に戻ります.

15.7 固定ディスクボリューム初期化

固定ディスクのボリューム初期化では物理フォーマッティングと論理フォーマッティングを行います．物理フォーマッティングは工場出荷時の不良トラックの検査，ボリュームID部の設定（スキュー），データ部の初期化，トラックの読み書き検査を行います．論理フォーマッティングはVOLラベル，ディレクトリ，ID，FAT等の作成を行います．これらの事を行うために，「固定ディスク初期化プログラム**format.hd**」が用意されています．format.hdは他のOS（例えばMS-DOS）が使用している領域を破壊しないようにBASICの領域を確保していきます．

以下，順を追ってこのプログラムの使い方を説明します．

(1) **"format.hd"** をロードします．

load "format.hd" [↵]

(2) プログラムを実行させます．

[f·5] あるいは **run** [↵]

(3) すると画面に次のメッセージが表示されます．

フォーマットするドライブは

初期化を行う固定ディスクのドライブ番号を指定して下さい．たとえば，3 [↵] と入力します．

(4) 指定したドライブ番号のディスク容量が10Mバイトの固定ディスクドライブであったならば，

ディスクタイプ＝10MB n1 (n2) メガバイトが使用可能です

と表示されます．ここでn1，n2は次のような意味を持っています．

n 1 ：使用可能な領域のサイズを意味します（固定ディスク全体のサイズからBASIC以外のOSが使用している領域のサイズを引いた値です）．

n 2 ：n1のうち，BASICで使用している領域のサイズを意味します．

初めてフォーマットする場合，次のように表示されます．

ディスクタイプ＝10MB 10 (0) メガバイトが使用可能です

また，BASICで3MB，MS-DOSで5MBすでにフォーマット済みの場合，次のように表示されます．

ディスクタイプ＝10MB 5 (3) メガバイトが使用可能です

この場合，どのOSでも使用されていない未使用領域のサイズは2MBです．

その後，続けてBASICのために確保する容量をたずねてきます．

フォーマットするサイズは（メガバイト）？

使用した領域のサイズをメガバイト単位で応えます．（この例の場合，2Mバイト分の領域が確保されます）．

他のOSで使用している領域もBASICで使用したい場合，現在の未使用サイズよりも大きな値を指定します（他のOSの領域は全てBASICのために解放されます）．この場合，(5)**物理フォーマッティングを行いますか（y/n）？**に対してはyを指定して下さい．nを指定するとフォーマットは行わず，自動的に(4)に戻ります．

(5) 次の問い合わせがあります．

物理フォーマッティングを行いますか（y/n）？

物理フォーマッテイングを行う場合はyと応答して下さい．物理フォーマッテイング済みで必要がない場合はnと応答して下さい．

物理フォーマッテイングを行いますと，他のOSの領域も全てクリアーされてしまいます．

(6) (5)でyと応答すると

ディスクの全領域をフォーマットします

確認しましたか（y/n）？

という表示が出ます．

確認した結果が正しければ“y”と応答して下さい．すると，

ドライブ3をフォーマット中です

と表示し，物理フォーマッティングを行います．

10MB固定ディスクの場合で約5分必要です．横棒グラフで処理進行状況を表示します．次に(7)に進みます．

(5)でnと応答すると(7)の処理に入ります．

(7) 論理フォーマッティングを開始すると

処理中です

と表示します．

(8) 正常に終了すると

終了しました

が表示されます．

注　意　固定ディスクへDISK code, IPLおよび日本語辞書ファイルをコピーする場合，BASICの領域が固定ディスク内の先頭に位置していないと他のOS（たとえば，MS-DOS）の領域を破壊してしまいます．したがって，固定ディスクからBASICを立ち上げるためにDISK codeとIPLの設定をおこなう場合，他のOSのためのフォーマッティングをおこなう前に必ずBASICのためのフォーマッティングをおこなってください．

15.8　固定ディスク障害ボリューム・ファイル復旧

固定ディスクのボリュームまたはファイルを使用中に，使用しようとした領域に障害がある場合には“Disk I/O error”メッセージが表示されます．この場合には今まで作成したボリュームまたはファイルの領域をできるだけ多く復旧する必要があります．すなわち，領域の管理単位はクラスタです．固定ディスクの場合は1クラスタは1トラック（33セクタ）からなっています．読み書きの単位は1セクタ（256バイト）で，障害検出もこの単位で行われます．それゆえ，1クラスタ中の1セクタでも障害があると，このクラスタ全体が障害のあるクラスタとみなされ，他の正常なセクタも共に使用できなくなります．このような正常なセクタをできるだけ復旧することが必要です．このためのプログラムが「固定ディスク障害ボリューム・ファイル復旧プログラム（recov.hd)」です．

復旧する対象領域をボリューム全体とするか，当該ファイル領域とするかによって「ボリューム復旧」，または「ファイル復旧」とよび，使用上の区別を行います．また，実際に復旧するというの

は障害を起こしたセクタをもつクラスタを障害のない正常なセクタだけからなるクラスタにできるだけ多くのセクタをコピーし，入れ換えを行うことです．システム領域についても対象とします．

以下，順を追ってこのプログラムの使い方を説明します．

(1) **"recov.hd"** をロードします．

load "recov.hd" [↵]

(2) プログラムを実行させます．

[f·5] あるいは **run** [↵]

(3) すると画面に次のメッセージが表示されます．

ファイル単位ですか，ボリューム単位ですか（f/v）？

ファイル復旧（f）を行うか，ボリューム復旧（v）を行うかを指定します．

(4) (3)でファイル復旧（f）を指定すると，当該ファイル名を問い合わせてきます．

ファイル名は？

ドライブ番号（d:），ファイル名（ffffffff 9桁まで）によって応答します．

"d : ffffffff"

復旧処理が始まります．

(5) (4)を繰り返すかどうかを問い合わせてきます．

処理を繰り返しますか（y/n）？

繰り返す場合はy，終了する場合はnで応答します．yの場合は(4)が繰り返されます．

(6) (3)でボリューム復旧(v)を指定すると，当該ボリュームのドライブ番号を問い合わせてきます．

ドライブ番号は？

障害ボリュームのドライブ番号を入力してください．たとえば，3 [↵] というように入力してください．

注 意 代替しようとする未使用の正常なクラスタが無い場合に次のメッセージが表示されます．

復旧処理ができません

15.9 固定ディスクファイルディレクトリ表示プログラム

固定ディスクに格納されているファイルを管理するためのユーティリティプログラムです．

① ディレクトリ表示機能

格納されているファイルをユーザ識別名ごとに整理して，ファイルディレクトリを表示または出力します．

② ファイル削除機能

ユーザ識別名とファイル名を指定してファイルを削除します．

これらの事を行うために，「固定ディスクファイルディレクトリ表示プログラム（**dir.hd**）」が用意されています．

以下，順を追ってこのプログラムの使い方を説明します．

(1) **"dir.hd"** をロードします．

load "dir.hd" [↵]

(2) プログラムを実行させます.

[f･5] あるいは run [↵]

(3) すると画面に次のメッセージが表示されます.

ファイルディレクトリ表示

ドライブ番号は?

ファイルディレクトリを表示したい固定ディスクのドライブ番号を入力します.

(4) どの機能を行うのかをプログラムからたずねてきます.

機能(1. 表示　2. 削除)?

数字1または2で応えてください.

1なら表示, 2なら削除

(5) さらに, 次のメッセージによる問い合わせがあります.

ユーザ識別名(XXX/999/000)?

XXX:特定ユーザのファイルを対象とするもので, ユーザ識別名3桁を指定

999:共通ファイルを対象とするときは999

000:全利用者のファイルを対象とするときは000

のいずれかをキーインしてください.

(4)で1と応答した場合には(6)へ進みます.

(4)で2と応答した場合には(9)へ進みます.

(6) 画面に表示するか, プリンタに印字するかをたずねてきます.

出力装置は(1. CRT　2. PRINTER)?

1または2の数字のいずれかで応えてください.

(7) (5)で指定したユーザ識別名のファイルが表示または印刷されます.

(8) 表示が終了すると, 次のメッセージが表示されます.

次の処理をしますか(y/n)?

処理を繰り返すかどうかたずねてきます.

yなら繰り返しです. (4)へもどります.

nならプログラムの実行を終ります.

(9) (4)で2と応えた場合は, 削除するファイル名をたずねてきます.

削除したいファイル名は?

ファイル名を入力してください.

処理が終了すると(8)へ行きます.

15.10 固定ディスク退避/復旧処理

「15.4 フロッピィディスク上のファイル転送」で説明した固定ディスクファイル転送プログラムは固定ディスク内のファイルをフロッピィディスクに転送する場合, 1枚のフロッピィディスクに格納できる範囲でしか転送はおこなわれません.

一枚のフロッピィディスクで納まらないような大容量のファイルをフロッピィディスクに退避させたい場合“backup.hd”を使用します．“backup.hd”により複数のフロッピィディスクに退避したファイルを固定ディスクに復旧することもできます．

なお，“backup.hd”でフロッピィディスク間のファイル転送をおこなうことはできません．

その手順は次のようになっています．

(1) “backup.hd”をロードします．

　　load “backup.hd” [↵]

(2) プログラムを実行させます．

　　[f･5] あるいは run [↵]

(3) 画面に次のメッセージが表示されます．

　　ファイルのバックアップを行います

　　送り側のドライブ番号は？

送り元の装置のドライブ番号を問い合わせてきます．

固定ディスク内のファイルをフロッピィディスクに退避させたい場合，固定ディスク装置のドライブ番号を指定します．

また，フロッピィディスクに退避したファイルを固定ディスクに復旧させたい場合，フロッピィディスク装置のドライブ番号を指定します．

(4) 送り先の装置のドライブ番号を問い合わせてきます．

　　受け側のドライブ番号は？

送り元が固定ディスクの場合，フロッピィディスク装置のドライブ番号を指定します．

また，送り元がフロッピィディスクの場合，固定ディスク装置のドライブ番号を指定します．

(5) 処理の対象とすべきファイル名を問い合わせてきます．

　　ファイル名は？

(6) ファイル名を入力します．

　　ファイル名は？　ffffffff [↵]

(7) フロッピィディスクの挿入要求メッセージが表示されます．

　　ドライブXXにnn枚目のフロッピィディスクをセットして下さい

nnは何枚目のフロッピィディスクであるかを示す数値です（最初は01です）

(8) 1枚目のフロッピィディスクをセットしたら [↵] キーを入力します．

(9) 次のメッセージが表示されファイルの退避/復旧処理が始ります．

　　ffffffff　　m−>n

ffffffffは処理中のファイル名です．

mは送り元の装置のドライブ番号です．

nは送り先の装置のドライブ番号です．

(10) 固定ディスク内のファイルのファイル容量が1枚のフロッピィディスクの容量を越える場合，退避処理中に次のメッセージが表示されます．

　　ドライブXXにnn枚目のフロッピィディスクをセットして下さい

nnは何枚目のフロッピィディスクであるかを示す数値です．

フロッピィディスクを掛け替え，[↵] キーを入力すると退避処理が継続されます．

同様に，複数のフロッピィディスクに退避されているファイルを固定ディスクに復旧する場合も，1枚目のフロッピィディスクの処理が終了すると，上記メッセージが表示されます．

この場合もフロッピィディスクを掛け替え ⏎ キーを入力してください

⑾ 退避/復旧処理が終了すると次のメッセージが表示され，他に処理すべきファイルがあるかどうか問い合わせてきます．

別のファイルを転送しますか（y/n）？

さらに処理すべきファイルがある場合y ⏎ と入力します．

(5)にもどり新たなファイルの処理を要求することができます．

n ⏎ と入力すると**終了しました**が表示され，プログラムは終了します．

終了しました

注　意　**"backup.hd"** プログラムで作成した，複数フロッピィディスクにまたがるファイルをBASICで直接扱うことはできません．

15.11　利用者定義文字格納ファイルの作成・更新

BASICは，システム立ち上げ時に，システムディスク内に **"usfontn88"** というファイルがあると，そのファイルより利用者定義文字のパターンを取出し，本体内の専用RAMに設定します．

この，**"usfontn88"** という利用者定義文字の格納ファイルを作成したり，更新したりするには，システムディスクに入っている **"mkfont.n88"** というプログラムを使います．

その手順は次の様になっています．

(1) 利用者定義文字格納ファイル作成，更新プログラム **"mkfont.n88"** をロードします．

load "mkfont.n88" ⏎

(2) プログラムを実行させます．

f·5 または **run** ⏎ ．

注　意　このプログラムでは，利用者が作成した文字パターンを表示するために，高解像のグラフィック画面を使用します．このため，専用高解像度ディスプレイが必要です．

(3) タイトルを表示した後で，次のメッセージが表示されます．

ドライブ番号は（1－X）？

Xはドライブ番号の最大値です．

(4) 利用者定義文字格納ファイルの作成，更新を行うシステムディスクがセットされているドライブ番号を入力します．

ドライブ番号が1の場合ですと，

1 ⏎

と入力します．

(5) すると，次のメッセージが出力されます．

(i) システムディスクにファイル **"usfontn88"** がない場合，または，ファイル **"usfontn88"** のなかに，利用者定義文字が1つも登録されていない場合，

登録済み漢字コード：ありません

機能（0．終了　1．追加）

(ii) システムディスクのファイル“usfontn88”に利用者定義文字が登録されており，さらに新たな登録が可能な場合，

登録済み漢字コード：

XXXX XXXX…XXXX

機能（0．終了　1．追加　2．更新　3．削除　4．表示）

(iii) システムディスクのファイル“usfontn88”に利用者定義文字が全て登録されている場合，

登録済み漢字コード：

XXXX XXXX…XXXX

機能（0．終了　1．更新　2．削除　3．表示）

なお，XXXXは，16進形式の登録済み漢字コードです．

(6) 終了，追加，更新，削除，表示の中から，適当な動作モードを選んでください．たとえば，(5)の(ii)の場合で更新を行うときには，

2 ⏎

と入力します．

終了を入力した場合には，システムディスクにファイル“usfontn88”を作成して終了します．

(7) (6)で更新，削除，表示を入力した場合には，

XXモード（0．1文字　1．全部）

と表示されます．ここでXXは，それぞれ更新，削除，表示です．

登録されている利用者定義文字の特定の1文字を，更新，削除または表示する場合には，

0 ⏎

と入力して下さい．すると(8)へ進みます．

登録されている利用者定義文字の全てを更新，削除または表示する場合には，

1 ⏎

と入力して下さい．すると，削除の場合は，(9)へ，更新，表示の場合は(10)へ進みます．

(8) (6)で追加を入力した場合，または(7)で0を入力した場合，次のメッセージが表示されます．

漢字コードは（7621−767E hex OR 7721−777E hex）：

利用者定義文字に対応する漢字コードは，16進形式で7621から767Eおよび7721から777Eまでですので，追加，更新，削除または表示する利用者定義文字の漢字コードを16進形式で入力して下さい．たとえば，漢字コードが762Aのときには，

762A ⏎

と入力します．すると，削除の場合は，(9)へ，その他の場合は，(11)へ進みます．

(9) 利用者定義文字の全てを削除する場合には，

利用者定義文字全部を削除しますか（y/n）？

特定の1文字を削除する場合には，

利用者定義文字XXXXを削除しますか（y/n）？

と確認してきます．ここで，XXXXは(8)で入力した漢字コードです．

もし，まちがっていれば，

n ⏎

と入力します．正しければ，

y ⏎

と入力します.

すると，再び(5)に戻ります.

(10) 追加または更新の場合には，それぞれ，

利用者定義文字を作成します

または，

利用者定義文字を更新します

と表示します.

次に，

漢字コード=××××（××××は16進形式漢字コード）

を表示すると同時に，利用者が定義した文字パターン（横16ドット×縦16ドット）を，次の様に表示します.

(line)	(hex)	(dot image)
1	××××	○○○○○○○○○○○○○○○○
2	××××	○○○○○○○○○○○○○○○○
≀	≀	≀
16	××××	○○○○○○○○○○○○○○○○
	16進形式	文字パターンを“・”と“*”で表示する.（“・”は0，“*”は1に対応）

但し，追加の場合には，16進形式は全て0，文字パターンは全て“・”です.

追加，更新の場合には，表示後，16進形式の表示の左上の文字のところにカーソルが表示されますから，以下のキーを操作して，文字パターンの作成，変更を行って下さい．表示の場合には，ただちに(12)へ進みます.

[←] [↑] [→] [↓] キー

カーソルが移動します.

[0]～[9]，[A]～[F] キー

カーソルが16進形式表示の位置にあるときのみ有効で，その位置の値を入力キーの値にします．同時に対応する文字パターンの部分も変化します．その後，カーソルは右へ移ります.

[ ]（スペースキー），[*] キー

カーソルが文字パターン表示の位置にあるときのみ有効で，その位置の値を“・”あるいは“*”にします．同時に対応する16進形式の部分も変化します.

その後，カーソルは右へ移ります.（文字パターンの右端では，カーソルは変化しません.）

[↵] キー

カーソルが一番下の行にないときには，次の行の16進形式の先頭文字にカーソルが移ります.

一番下の行にカーソルがある場合には，文字パターンの作成，変更が終了したものとして(12)へ進みます.

ESC キー

カーソルの位置とは関係なく，文字パターンの作成，変更が終了したものとして，⑾へ進みます．

⑾ 文字パターンの作成，更新または表示が終了すると文字パターンを表示している右側に

文字パターン　■

と表示します．■は，日本語文字として，実際に使用される場合の文字イメージです．

さらに，追加，更新の場合には，

よろしいですか（y/n）？

を，表示の場合には，

リターンキーを押して下さい

を表示します．

追加または更新で，文字パターンがまちがっているときには，

n ↵

と入力します．⑽で行った作成，変更はすべて無効となりますので，再び⑽の始めからやり直して下さい．

追加または更新が正しく終了した場合には，

y ↵

と入力します．

表示の場合には，↵ キーのみを入力します．

その後，追加の場合または，⑺で0を入力した場合には，⑸へ戻ります．更新，表示の場合で，⑺で1を入力した場合には，登録されている利用者定義文字のすべてに対して，⑽と⑾を繰返した後に，⑸へ戻ります．

NOTE ⑹で終了を入力する前に，STOP キーを入力すると，

処理が中断されました．処理を中止しますか（y/n）？

のメッセージを表示します．そこで，

y ↵

を入力すると，システムディスクの“usfontn88”ファイルはもとのまま，終了します．一方，

n ↵

を入力すると，再び⑶からやり直すことが出来ます．（ただし，いままでの入力はすべて無効となります．）

また，処理の途中で何らかの異常を検出した場合も同様です．

15.12　メモリスイッチの設定

PC-9800シリーズのシステムは，本体に8バイトの不揮発性メモリを持っており，このメモリは電源が断たれても，その状態を保持しています．PC-9800シリーズでは，このメモリスイッチに，ユーザーシステムの環境に合わせた情報を持っており，このスイッチの内容を変更することによって，システム立ち上げ時の状態をあらかじめ設定できます．

switch.n88は，このメモリスイッチを変更するユーティリティです．

備　考　メモリスイッチは，BASICの機械語モニタのセットメモリスイッチ機能により直接変更することができます．
この方法については，「第21章　メモリスイッチ」を参照してください．

以下，メモリサイズを変更する場合を例にその手順を示します．

① メモリスイッチ設定ユーティリティをロードします．

load "switch.n88" ⏎

② プログラムを実行させます．

f·5 または run ⏎

③ 次のメッセージが表示されます．

***** switch set. MENU *****

RS-232C（初期設定）
RS-232C（送受信コード）
メモリサイズ
数値演算プロセッサ
CRT　初期画面　色指定（TEXT）
拡張　ボード
画面ハードコピー
立ち上げ装置
Basic mode
Network Basic
終了
↑　↓　←　→ ……他を選択します．
⏎　……決定します．

注　意　上記のメッセージは，専用高解像度ディスプレイが接続されている場合のみ表示されます．その他のシステムでは，上記のメッセージに相当するものが英字で表示されます．

④ ↓ キーを2回押し，そこで ⏎ キーを押すと，次のメッセージが表示されます．

……メモリサイズ……
サイズ・・・・byte…………………………………………………………
128K　256K　384K　512K　(640K)

(640K) は，現在のメモリの設定が，640Kbyteになっていることを示しています．

⑤ ここで ← キーを押し，実装されているメモリサイズを選択します．

……メモリサイズ……
サイズ・・・・byte…………………………………………………………
128K　256K　384K　512K　(640K)

ここで ⏎ キーを押すと

……メモリサイズ……
サイズ・・・・byte…………………………………………………………
128K　256K　384K　(512K)　640K

上記のように，メモリサイズが決定します．

⑥　[ESC] キーを押し，初期画面に戻ります．

＊＊＊＊＊ switch set. MENU ＊＊＊＊＊

RS-232C（初期設定）
RS-232C（送受信コード）
[メモリサイズ]
数値演算プロセッサ
CRT 初期画面　色指定（TEXT）
拡張　ボード
画面ハードコピー
立ち上げ装置
Basic mode
Network Basic
終了

⑦　[↓] キーを数回押し，“終了”を選択し，[⏎] キーを押すことにより本ユーティリティは終了します．

最後に次のメッセージが表示されます．

メモリスイッチを設定し　終了しました． スイッチの内容を変更した時は，リセットスイッチを押して再立ち上げしてください．

注　意

(1)　本ユーティリティ実行中に [STOP] キーを押すことにより，中途終了することはできますが，その場合メモリスイッチの設定はおこなわれません．

(2)　本ユーティリティ実行中の，状態及び各キーは次のような対応になっています．

（　）　…………現在の値を示します
ESC　…………始めの画面に戻ります
HELP　…………説明をします
↑　↓　←　→　…………他を選択します．
⏎　…………決定します．

(3)　本ユーティリティによってメモリスイッチの内容を変更した場合は，本体のディップスイッチSW2の5を必ずONにして下さい．

そして，必ずシステムを再立ち上げして下さい．

(4)　本ユーティリティによって設定できる，項目は以下のとおりです．

```
……RS-232C（初期設定）……
  Xパラメータ…………………………………………………………………
              （無効）          有効
  通信方法……………………………………………………………………
              （全二重）        半二重
  データbit長………………………………………………………………
              （7 bit）         8 bit
  パリティ　チェック…………………………………………………
              （使わない）奇数　偶数
  ストップbit長……………………………………………………………
              （1 bit）    1.5bit     2 bit
  ボーレート…………………………………………………………………
     75        150       300       600
     （1200）  2400      4800      9600
```

```
……RS-232C（送受信コード）……
  日本語シフトコード
              （KI=1B4Bh  KO=1B48h）
               KI=1A70h  KO=1A71h
  CR受信処理…………………………………………………………………
              （CR　復帰＋改行）
               CR・LF　復帰＋改行　CR復帰
  リターンキー押下時の送信処理……………………………………
              （CR）            CR・LF
  Sパラメータ………………………………………………………………
              （無効）          有効
  DELコード受信時動作（ターミナルモード/入出力モード）‥
              （BS/DEL）        NUL/NUL
```

……メモリサイズ……

サイズ……byte……………………………………………………………………

128K　256K　384K　512K　(640K)

……数値演算プロセッサ……

V30用数値演算プロセッサ……………………………………………………

(使わない)　使う

……CRT初期画面　色指定（TEXT）……

色…………………………………………………………………………………

(白)　緑

……拡張　ボード……

サウンドボード………………………………………………………………

(使わない)　使う

RS-232C（第2回線/第3回線）ボード…………………………………

(使わない)　使う

GP-IBボード…………………………………………………………………

(使わない)　使う

……画面ハードコピー……

PC-PR201系プリンタ……………………………………………………

使わない　(使う)

拡張画面ハードコピー機能………………………………………………

(使わない)　使う

カラーコピー…………………………………………………………………

(モノクロコピー)　カラーコピー

……立ち上げ装置……

立ち上げ装置…………………………………………………………………

(フロッピィディスク－＞ hard disk)

BOOT on 640KBフロッピィディスク

BOOT on 1MBフロッピィディスク

BOOT on hard disk 1

BOOT on hard disk 2

ROM BASIC

固定ディスク優先割付け…………………………………………………

(指定しない：floppy disk －＞ hard disk)

指定する　：hard disk －＞ floppy disk

固定ディスクユーザ識別名………………………………………………

(使う)　使わない

```
……Basic mode………
  電話制御命令の使用…………………………………………………………………
                    (使わない)        使う
  拡張モニタモード……………………………………………………………………
                    (使わない)        使う
```

```
…Network Basic……
  ネットワークBasic…………………………………………………………
                    (使わない)        使う
  ドライブ番号割付け  *network*……………………………………………
                    (仮想ドライブ－〉  ローカルドライブ)
                    ローカルドライブ－〉  仮想ドライブ
```

15.13　文節辞書ファイルの保守

文節辞書ファイル保守プログラムとして“dicmen.n88”が用意されています．

“dicmen.n88”は次の3つの機能を持っています．

(1)　文節辞書ファイルに対する単語の登録

(2)　文節辞書ファイルに登録されている利用者登録単語の削除

(3)　文節辞書ファイルに登録されている単語の一覧表出力

(1)　“dicmen.n88”をロードします．

　　load “dicmen.n88” [↵]

(2)　プログラムを実行させます．

　　[f･5] または run [↵]

(3)　次のメッセージが出力されます．

　　辞書ファイルのドライブ番号を入力して下さい

　辞書ファイルが格納されている媒体のドライブ番号を入力します．

　ドライブ番号が1の場合

　　1 [↵]

　と入力します．

(4)　次に辞書ファイル名の問い合わせがおこなわれます．

　　辞書ファイルの名前を入力して下さい．

　たとえば次のように応答します．

　　BUNSET.SU [↵]

(5) 次に示すメニュー画面が表示されます．

```
↑↓でカーソルを移動して機能を選択して下さい．
  1  単語の登録
  2  単語の削除
  3  単語の一覧
  4  終了
```

任意の機能項目にカーソルを位置付けて[↵]キーを押してください．

(6) **単語の登録**を選んだ場合，次のメッセージが表示されます．
かな見出し（単語の読み）を入力します．

```
                    〈単語の登録〉
カナ見出しを入力して下さい．
カナ見出し■
```

登録したい単語のカナ見出しを日本語で入力します．日本語入力の方法は，その時のシステムの日本語入力環境に依存します．以降の日本語入力に関しても同様です．

カナ見出しは16文字以内の英数，カナ，記号（“＊”，“！”を除く）の組み合わせです．

ここで何も入力しないで[↵]キーを押すとメニュー画面にもどります．

カナ見出しの入力に誤りがある時には，ブザーが鳴り，再びカナ見出しの入力に戻ります．

ここでは**NEC**というカナ見出しを入力したことにします．

(7) カナ見出しの入力が終ると，次に登録したい単語を入力します．

```
                    〈単語の登録〉
単語を入力して下さい．
カナ見出し    NEC
単語          ■
```

単語は16文字以内の日本語で入力します．

ここで，何も入力しないで[↵]キーを押すと，カナ見出しの入力に戻ります．

単語の入力に誤りがあるときにはブザーが鳴り再び単語の入力に戻ります．

ここでは**日本電気株式会社**と入力したことにします．

(8) 次に品詞の指定をおこないます．

```
                    〈単語の登録〉
品詞の指定をして下さい.
カナ見出し        NEC
単語              日本電気株式会社
品詞の指定をしますか（Y/N）？■
```

品詞の指定をしない場合Nと入力します.
すぐに単語の登録がおこなわれ，カナ見出しの入力にもどります.
品詞の指定をおこなう場合Yと入力します.

(9)　品詞の一覧が表示されます.

```
                    〈単語の登録〉
←，→キーでカーソルを移動して品詞を選択してください.
カナ見出し        NEC
単語              日本電気株式会社
品詞              基本語　動詞　固有名詞
```

カーソルを移動し，任意の品詞に位置付け，↵キーを入力します.

(10)　次に品詞の詳細指定をおこないます.
たとえば固有名詞を選んだ場合，次のように表示されます.

```
←，→，↑，↓キーでカーソルを移動して品詞を選択して下さい.
カナ見出し        NEC
単語              日本電気株式会社
品詞              基本語　　動詞　　固有名詞
                  苗字　　名前　　地名　　団体名
                  会社名　　建物名　　商品名
```

カーソルを移動し，任意の品詞に位置付け，↵キーを押して下さい.
ここでは会社名を選んだことにします.

(11)　最後に登録単語の確認をおこないます.

〈単語の登録〉

確認して下さい.

カナ見出し　NEC

単語　日本電気株式会社

品詞　基本語　動詞　固有名詞

　苗字　名前　地名　団体名

　会社名　建物名　商品名

よろしいですか　(Y/N)　?　■

Yを入力すると単語の登録がおこなわれます.

Nを入力するとカナ見出しの入力に戻ります.

なお，重複した登録（カナ見出し（読み），対応する単語，品詞のすべてが一致するもの）は許されません.

この場合，ブザーと共に次のメッセージが表示され，カナ見出しの入力に戻ります.

すでに登録されています.

(12) 単語の登録が完了すると,

〈単語の登録〉

単語を登録しました.

カナ見出し　NEC

単語　日本電気株式会社

品詞　基本語　動詞　固有名詞

　苗字　名前　地名　団体名

　会社名　建物名　商品名

終りますか (Y/N)　?　■

と表示されます.

Yを入力すると機能選択のメニュー画面に戻ります.

Nを入力すると，カナ見出しの入力に戻ります.

備　考　(1) 品詞として基本語を選択した場合，品詞の詳細は次の中から選びます.

名詞　サ変名詞注(1)　形容詞　形容動詞

副詞　接続詞

注（1）　サ変名詞は“する”をつけることができる名詞，たとえば解析（する），感動（する）等を指します.

(2) 固有名詞およびサ変名詞と接続可能な語句（接尾語）の関係は次のとおりです.

〈接尾語〉

・A……論，率，力，料，流，名，法，別，文，物，部，病，日，品，表，費，内，等，的，片，中，値，高，代，説，性，上，所，時，室，書，者，策，後，権，組，業，金，局，級，型，額，後，会，化，員，案

・B……用
・C……風，港，駅
・D……殿
・E……式
・F……氏，君
・G……様

	A	B	C	D	E	F	G
名　　詞	○	○	○	○	○	×	○
団体名	×	×	×	○	○	×	○
商品名	×	×	×	×	○	×	×
地　　名	×	×	○	×	○	×	×
サ変名詞	×	×	×	×	×	×	×
会社名	×	×	×	○	○	×	○
苗　　字	×	○	×	○	○	○	○
名　　前	×	○	×	○	○	○	○
建物名	×	×	×	×	○	×	×

○→接続可
×→接続不可
（この表の見方）
商品名に続けることができるのはE(式)のみです
たとえば，NECというカナ見出しに対し“日本電気”を商品名として登録しておきますと日本語入力の際，“NECしき”という読みの入力で日本電気式という文節変換が可能になります。

(3) 品詞として動詞を選んだ場合，品詞の詳細は次の中から選びます．

カ行５段　ガ行５段　サ行５段　タ行５段　ナ行５段　バ行５段
マ行５段　ラ行５段　アワ行５段　１段　サ行変格　カ行変格

それぞれの意味は次の通りです．
５段活用……口語について語尾が五十音図のア列，イ列，ウ列，エ列の４段に活用するもの

例	基本形	未然	連用	終止	連体	已然	命令
カ行５段	行く	か（こ）	き	く	く	け	け
ガ行５段	泳ぐ	が（ご）	ぎ	ぐ	ぐ	げ	げ
サ行５段	押す	さ（そ）	し	す	す	せ	せ
タ行５段	打つ	た（と）	ち	つ	つ	て	て
ナ行５段	死ぬ	な（の）	に	ぬ	ぬ	ね	ね
バ行５段	飛ぶ	ば（ぼ）	び	ぶ	ぶ	べ	べ
マ行５段	飲む	ま（も）	み	む	む	め	め
ラ行５段	乗る	ら（ろ）	り	る	る	れ	れ
アワ行５段	思う	わ（お）	い	う	う	え	え

カ変活用……文語「こ・き・く・くる・くれ・こ」，口語「こ・き・くる・くる・くれ・こい」と活用するもの．文語「来（く)」，口語「来（く）る」の各一語．

サ変活用……文語「せ・し・す・する・すれ・せよ」，口語「し（せ・さ）・し・する・すれ・しろ（せよ)」と活用するもの．

例	基本形	未然	連用	終止	連体	已然	命令
文語	為す	せ	し	す	する	すれ	せよ
口語	為す	せ（し）	し	する	する	すれ	せよ（しろ）

一段活用……語尾が五十音図のイ列（上一段)，又はエ列（下一段）の一段だけに活用するもの．

例	基本形	未然	連用	終止	連体	已然	命令
上一段	着る	き	き	きる	きる	きれ	きよ（ろ）
下一段	得る	え	え	える	える	えれ	えよ（ろ）

(13)　(5)で**単語の削除**を選んだ場合，次のメッセージが表示されます．

> **〈単語の削除〉**
> **カナ見出しを入力して下さい．**
> **カナ見出し■**

カナ見出しを**単語の登録**の場合と同じ方法で入力します．

ここではNECを入力したとします．

もし対応する利用者登録単語が存在しない場合，次のメッセージが表示されるとともにブザーが鳴り，再びカナ見出しの入力に戻ります．

指定された利用者登録単語は存在しません．

(14)　指定した単語が表示されますので，その中から削除する単語を選びます．

> **〈単語の削除〉**
> **↑，↓キーで単語を選択して下さい．**
> **カナ見出し**　　NEC
> **登録単語**　　日本電気㈱ 固・会社

削除したい単語を選び，[↵]キーを押します．

(15)　次の確認メッセージが表示されます．

> **〈単語の削除〉**
> **確認して下さい**
> **カナ見出し**　　NEC
> **登録単語**　　日本電気㈱ 固・会社
> **よろしいですか（Y/N）？　■**

Yを入力すると，選択した単語が削除されます．

Nを入力すると，カナ見出しの入力にもどります．

⒃ 単語の削除が完了すると次のメッセージが表示されます．

〈単語の削除〉

単語を削除しました．
カナ見出し　　NEC
登録単語　　　日本電気㈱/固・会社
終りますか（Y/N）？ ■

Yを入力すると，機能選択のメニュー画面にもどります．

Nを入力すると，カナ見出しの入力にもどります．

備　考 登録単語の品詞情報の意味は次の通りです．

/　　　……品詞指定無し	/基・接続……基本語/接続詞
/動・カ５……動詞/カ行５段活用	/基・副詞……　〃　/副詞
/動・ガ５……〃　/ガ　　〃	/基・形動……　〃　/形容詞
/動・サ５……〃　/サ　　〃	/基・形容……　〃　/形容動詞
/動・タ５……〃　/タ　　〃	/基・サ変……　〃　/サ変名詞
/動・ナ５……〃　/ナ　　〃	/基・名詞……　〃　/名詞
/動・バ５……〃　/バ　　〃	/固・商品……固有名詞/商品名
/動・マ５……〃　/マ　　〃	/固・建物……　〃　　/建物名
/動・ラ５……〃　/ラ　　〃	/固・会社……　〃　　/会社名
/動・ア５……〃　/ア　　〃	/固・団体……　〃　　/団体名
/動・１段……〃　/１段活用	/固・地名……　〃　　/地名
/動・サ変……〃　/サ行変格活用	/固・名前……　〃　　/名前
/動・カ変……〃　/カ　　〃	/固・苗字……　〃　　/苗字

注　意 利用者登録単語以外の削除をおこなうことはできません．

この場合，次のメッセージが表示されます．

削除できません．

⒄ ⑸で**単語の一覧**を選んだ場合，次のメッセージが表示されます．

〈単語の一覧〉

出力開始位置のカナ見出しを入力して下さい．
出力開始カナ見出し■

出力範囲の指定をおこないます．

まず，先頭のカナ見出しを**単語の登録**の場合と同じ方法で入力します．

ここで［↵］キーのみを入力すると辞書の先頭からの出力となります．

ここでは**あい**と入れたことにします．

⒅ 開始位置を入力し終ると次のメッセージが表示されます．

〈単語の一覧〉
出力終了位置のカナ見出しを入力して下さい.
開始カナ見出し　　あい
終了カナ見出し　　■

出力したい最後のカナ見出しを入力します.

ここで ⏎ キーのみを入力すると辞書の最後までの出力となります.

(19) 次の形式で一覧表が出力されます.

アイ	哀	/基名詞
		/
	愛	/動サ5
		/動サ変
		/基名詞
	相	/基名詞
		/
アイイソ	相磯	/固苗字
アイイン	合印	/基名詞
:	:	:
:	:	:

なお，表示中の一時停止は CTRL ＋ S でおこないます.

(20) 最後に終りますか（Y/N）？のメッセージが表示されます.

Yを入力すると，機能選択メニュー画面にもどります.

Nを入力すると，開始カナ見出しの入力にもどります.

15.14 その他のユーティリティプログラム

「15.4 フロッピィディスク上のファイル転送」で説明した“xfiles.n88”を使用する場合は，ディスク媒体の種類にあったディスク装置が必要でした．ここでは，1MB/640KB装置を使用して，5インチ160KB/320KBの媒体上のファイルを読み出し1MB/640KBの媒体へファイル転送を行うプログラムについて説明します.

(1) ファイル転送プログラムをロードします.

load“DDconv.n88” ⏎

(2) プログラムを実行させます.

f·5 またはrun ⏎

(3) 次のメッセージが表示されます.

1D/2D媒体のファイルを2HD/2DD媒体に転送します

送り側のドライブ番号は？

(4)　このメッセージは，入力媒体（5インチ160KB/320KB媒体）をセットするドライブのドライブ番号を聞いているもので，たとえば，

　2 ⏎

と入力します．

(5)　さらに，次のメッセージが表示されます．

　ディスクタイプは（1．1D　2．2D）？

セットした媒体が1D（160KB）の場合には「1」を，2D（320KB）の場合には「2」を入力します．

　2 ⏎ （2Dの場合）

(6)　すると次のメッセージが表示されます．

　受け側のドライブ番号は？

(7)　このメッセージはファイルを転送する1MB/640KBの媒体をセットするドライブのドライブ番号を聞いているもので，たとえば，

　1 ⏎

と入力します．

なお，出力媒体となる1MB/640KB媒体はすでにフォーマッティングされていなければなりません．

(8)　次に

　確認しましたか（y/n）？

と確認の要求をしてきます．

　y ⏎

と入力します．

(9)　次のメッセージが表示されます．

　送り側ディスクをドライブ2にセットして下さい

　受け側ディスクをドライブ1にセットして下さい

入力媒体となる5インチ160KB/320KB媒体および出力媒体をセットします．セットが終ったら ⏎ キーを入力します．

次のようなメッセージが表示され，データの読み込みが始まります．

　ffffffffファイル転送中です　2－>1

　（ffffffffは処理中のファイル名です）

(10)　転送が終了すると，

　終了しました

と表示され，プログラムは終了します．

このプログラムは，ファイルを転送するだけですから，同じ名前のファイルがないかぎり，出力媒体に前から入っていたファイルを，消すことはありません．

また，DISK codeは転送しません．

第16章
マウス用ソフトウェアドライバ

マウスは，パーソナルコンピュータのディスプレイ上のカーソルの動きを容易にコントロールできるポインティングデバイスです．マウスを机上などで，自由に動かすことによってカーソルを操作者の希望の位置に動かすことができます．また，マウスには2つのボタンがついています．これらのボタンは押したり離したりすることができ，これによって，ソフトウェア（アプリケーションプログラムなど）とのやりとりが可能になります．例えば，カーソルをディスプレイ上のある場所（コマンドや記号など）に動かし，ボタンを押すと，ソフトウェアがそれを調べ，カーソルの位置するコマンドを実行します．この様にマウスは，キーボードに替わる入力装置として利用できます．

マウス用ソフトウェアドライバは，マウスを制御するプログラムです．

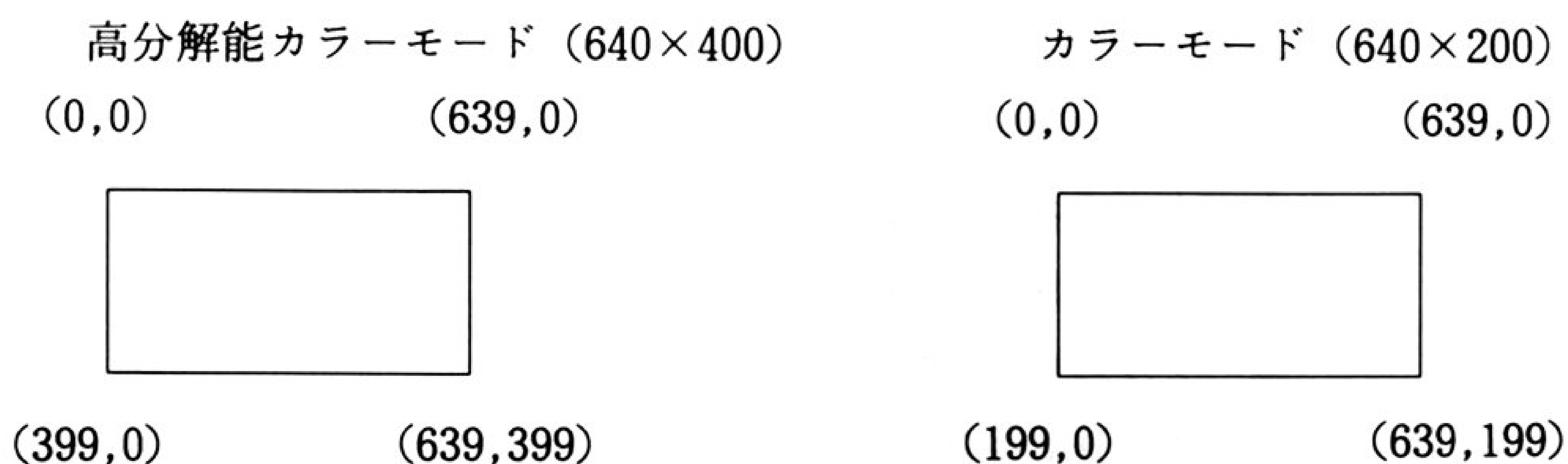
マウス用ソフトウェアドライバには，大きく分けて2つの機能があります．1つは，マウスから情報（マウスの移動距離やボタンの状態など）を取得して，カーソルを動かしたりする機能です．もう1つは，アプリケーションプログラムなどからの要求(ファンクションコール)により，カーソルの位置やマウスのボタンの状態などを通知したりする16個のファンクションを実行する機能です．つまり，マウス用ソフトウェアドライバはマウスとアプリケーションプログラムとの仲介をするものです．したがって，アプリケーションプログラムは，マウス用ソフトウェアドライバのファンクションを呼び出すことで，マウスの能力を最大限に活用できるわけです．

16.1　マウス用ソフトウェアドライバのための予備知識

ここでは，マウス用ソフトウェアドライバ（以降，ソフトウェアドライバと呼びます）を使用するにあたって必要となる情報について説明します．

(1)　画面の座標系

ソフトウェアドライバでは，接続されているCRTにより2種類の画面モードがあります．両画面モードともスクリーン上のドット位置を水平座標（横）と垂直座標（縦）とによって表現します．例えば，水平座標が5，垂直座標が10のドットでは（5，10）と表わします．



(2) 表示画面

ソフトウェアドライバは，画面モードによってスクリーンに表示する画面が異なります．高分解能カラーモードでは，次に示すグラフィック用VRAM(GVRAM0,GVRAM1,GVRAM2,GVRAM3)のそれぞれの開始アドレスから32キロバイトずつをGDC合成によってスクリーンに表示します．また，カラーモードでは，グラフィック用VRAMのそれぞれの開始アドレスから16キロバイトずつをGDC合成によってスクリーンに表示します．

グラフィック用VRAMの開始アドレス

GVRAM 0：（A8000)$_{16}$

GVRAM 1：（B0000)$_{16}$

GVRAM 2：（B8000)$_{16}$

GVRAM 3：（E0000)$_{16}$

(3) カーソル

カーソルは，画面モードによって大きさが異なり，高分解能カラーモードでは16×32ドット，カラーモードでは16×16ドットの大きさの四角形です．またカーソルには中心点というものがあります．これは，カーソルのスクリーン上の位置（座標）を決定するもので，カーソル内の1ドットがそれにあたります．カーソルの中心点は，カーソル内の左上角のドットを（0，0）とする座標系で表わします．

(4) ミッキー

マウスを動かした時の移動距離は，ミッキーという単位で表わします．1ミッキーは約100分の1インチです．ソフトウェアドライバでは，カーソルを8ドット移動するのに必要なマウスの移動距離を設定することで，マウスの感度を変えることができます．

16.2 ファンクション

ソフトウェアドライバは，アプリケーションプログラムの要求によって，以下の16種類のファンクションを実行することができます．

〔機能コード〕	〔機　能〕
0	初期化処理
1	カーソル表示
2	カーソル消去
3	カーソル位置の取得
4	カーソル位置の設定
5	左ボタンの押下情報の取得
6	左ボタンの解放情報の取得
7	右ボタンの押下情報の取得
8	右ボタンの解放情報の取得
9	カーソルの形の設定
11	マウスの移動距離の取得

12	ユーザー定義サブルーチンのコール条件の設定
15	ミッキー/ドット比の設定
16	水平方向のカーソル移動範囲の設定
17	垂直方向のカーソル移動範囲の設定
18	カーソルの表示画面の設定

機能コード10，13，14は存在しませんので御注意下さい．

ここでは，これらのファンクションの入力情報，出力情報とその説明をし，具体的な使用方法については，「16.3　マウス用ソフトウェアドライバの使用方法」で説明します．

○ファンクション0（マウス環境の初期化）

〔入　力〕

AX＝0（機能コード）

〔出　力〕

AX＝環境状態

0：マウスを使用できない環境

−1：マウスを使用できる環境

〔説　明〕

カーソルの表示，カーソルの形，カーソルの中心点，ミッキー/ドット比，マウスの割り込み周期等の環境を初期化します．

○ファンクション1（カーソル表示）

〔入　力〕

AX＝1（機能コード）

〔出　力〕

なし

〔説　明〕

カーソルをスクリーン上に表示させるためのファンクションです．1度このファンクションをコールすると，ファンクション名：カーソルの消去をコールするまで，カーソルがマウスの動きに従ってスクリーン上を動きます．

○ファンクション2（カーソル消去）

〔入　力〕

AX＝2（機能コード）

〔出　力〕

なし

〔説　明〕

スクリーン上に表示されているカーソルを表示させないためのファンクションです．カーソルは，1度消されるとファンクション1（カーソル表示）をコールするまで，スクリーンには表示さ

れません．しかし，カーソルは表示されていない間もマウスを動かすことによってカーソルはスクリーン上で移動しています．

○ファンクション3（カーソル位置の取得）

〔入　力〕

AX＝3（機能コード）

〔出　力〕

AX＝左ボタンの状態

0：離されている

－1：押されている

BX＝右ボタンの状態

0：離されている

－1：押されている

CX＝カーソルの位置の水平座標（0～639）

DX＝カーソルの位置の垂直座標

0～199：カラーモード

0～399：高分解能カラーモード

〔説　明〕

現在のカーソルの位置を得るためのファンクションです．カーソルの位置は，水平座標，垂直座標で得られ，それはカーソルの移動範囲内の値です．またこの時のマウスの右ボタン及び左ボタンの状態（押されているか，押されてない）も得ることができます．

○ファンクション4（カーソル位置の設定）

〔入　力〕

AX＝4（機能コード）

CX＝カーソルの新しい位置の水平座標（0～639）

DX＝カーソルの新しい位置の垂直座標

0～199：カラーモード

0～399：高分解能カラーモード

〔出　力〕

なし

〔説　明〕

カーソルの位置を指定の位置に設定するためのファンクションです．アプリケーションプログラムは，希望の位置の水平座標，垂直座標を指定して，本ファンクションをコールすると，カーソルはその位置に移動します．希望の位置がカーソル移動範囲外の場合には，移動範囲内の端にカーソルを移動します．

○ファンクション5（左ボタンの押下情報の取得）

〔入　力〕

AX＝5（機能コード）

〔出　力〕

AX＝左ボタンの状態

0：離されている

−1：押されている

BX＝左ボタンが押された回数

CX＝最後に左ボタンが押された時のカーソル位置の水平座標

DX＝最後に左ボタンが押された時のカーソル位置の垂直座標

〔説　明〕

マウスの左ボタンの押下（押されること）に関する各種の情報を取得するためのファンクションです．各種の情報とは，次のものです．

・左ボタンの現在の状態

・本ファンクションが最後にコールされてから，今回コールされるまでに左ボタンが押された回数

・最後に左ボタンが押された時のカーソルの位置（水平座標，垂直座標）

本ファンクションがコールされると，ソフトウェアドライバは以上の情報をアプリケーションプログラムに通知します．

○ファンクション6（左ボタンの解放情報の取得）

〔入　力〕

AX＝6（機能コード）

〔出　力〕

AX＝左ボタンの状態

0：離されている

−1：押されている

BX＝左ボタンが離された回数

CX＝最後に左ボタンが離された時のカーソル位置の水平座標

DX＝最後に左ボタンが離された時のカーソル位置の垂直座標

〔説　明〕

マウスの左ボタンの解放（離されること）に関する各種の情報を取得するためのファンクションです．各種の情報とは，次のものです．

・左ボタンの現在の状態

・本ファンクションが最後にコールされてから，今回コールされるまでに左ボタンが離された回数

・最後に左ボタンが離された時のカーソルの位置（水平座標，垂直座標）

本ファンクションがコールされると，ソフトウェアドライバは以上の情報をアプリケーションプログラムに通知します．

○ファンクション7（右ボタンの押下情報の取得）

〔入　力〕

AX＝7（機能コード）

〔出　力〕

AX＝右ボタンの状態

0：離されている

－1：押されている

BX＝右ボタンが押された回数

CX＝最後に右ボタンが押された時のカーソル位置の水平座標

DX＝最後に右ボタンが押された時のカーソル位置の垂直座標

〔説　明〕

マウスの右ボタンの押下に関する各種の情報を取得するためのファンクションです．各種の情報とは，次のものです．

・右ボタンの現在の状態

・本ファンクションが最後にコールされてから，今回コールされるまでに右ボタンが押された回数

・最後に右ボタンが押された時のカーソルの位置（水平座標，垂直座標）

本ファンクションがコールされると，ソフトウェアドライバは以上の情報をアプリケーションプログラムに通知します．

○ファンクション8（右ボタンの解放情報の取得）

〔入　力〕

AX＝8（機能コード）

〔出　力〕

AX＝右ボタンの状態

0：離されている

－1：押されている

BX＝右ボタンの離された回数

CX＝最後に右ボタンが離された時のカーソル位置の水平座標

DX＝最後に右ボタンが離された時のカーソル位置の垂直座標

〔説　明〕

マウスの右ボタンの解放に関する各種の情報を取得するためのファンクションです．各種の情報とは，次のものです．

・右ボタンの現在の状態

・本ファンクションが最後にコールされてから，今回コールされるまでに右ボタンが離された回数

・最後に右ボタンが離された時のカーソルの位置（水平座標，垂直座標）

本ファンクションがコールされると，ソフトウェアドライバは以上の情報をアプリケーションプログラムに通知します．

○ファンクション9（カーソルの形の設定）

〔入　力〕

AX＝9（機能コード）

BX＝カーソルの中心点の水平座標（0〜15）

CX＝カーソルの中心点の垂直座標

0〜15：カラーモード

0〜31：高分解能カラーモード

ES：DX＝カーソルの形を決定するデータのアドレス

データの形式は，カラーモードでは16×16ビット

高分解能カラーモードでは16×32ビット

〔出　力〕

なし

〔説　明〕

カーソルの形や中心点を設定するためのファンクションです．カーソルの形は，四角形のカーソルのうちのどのドットを表示するかによって決まります．つまり，表示されたドットの集合がカーソルとして見えるわけです．例えば，四角形のカーソルのドットを全て表示するように指定すると，カーソルの形は四角形になります．

〔例〕カーソルの形を決定するデータの形式

```
11000000B, 00000000B   ↑
11100000B, 00000000B
    :          :        カラーモードでは16行，
    :          :        高分解能カラーモードでは32行.
00000000B, 00000000B   ↓
```

また，カーソルの中心点は，カーソルの左上角のドットを（0，0）とした座標系の位置で与えます．

⋮

○ファンクション11（マウスの移動距離の取得）

〔入　力〕

AX＝11（機能コード）

〔出　力〕

CX＝マウスの水平方向の移動距離（−32768〜32767）

DX＝マウスの垂直方向の移動距離（−32768〜32767）

〔説　明〕

マウスの移動距離を取得するためのファンクションです．本ファンクションを最後にコールされた時のマウスの位置から，今回コールされた時点でのマウスの位置までの水平方向及び，垂直方向の相対的な距離をアプリケーションプログラムに通知します．水平方向では右の向きが正であり，垂直方向では，手前の向きが正となります．また距離の単位はミッキーで，その範囲は−32768から32767です．

○ファンクション12（ユーザー定義サブルーチンのコール条件の設定）

〔入　力〕

AX＝12（機能コード）

CX＝コール条件

ビット0：カーソル位置の変化

ビット1：左ボタンが押される

ビット2：左ボタンが離される

ビット3：右ボタンが押される

ビット4：右ボタンが離される

ビット5～15：未使用

（最下位ビットをビット0として，各ビットが1の時はコールし，0の時はコールしない）

ES：DX＝ユーザー定義サブルーチンのアドレス

〔出　力〕

なし

〔説　明〕

ユーザー（アプリケーションプログラム）が定義したサブルーチンをソフトウェアドライバがコールする条件と，そのサブルーチンのアドレスを設定するためのファンクションです．ソフトウェアドライバは，次の5つの事象が発生した時に，サブルーチンをコールすることができます．

・カーソルの位置が変わった場合
・左ボタンが押された場合
・左ボタンが離された場合
・右ボタンが押された場合
・右ボタンが離された場合

これらの現象のうちのどれが発生した時に，サブルーチンをコールするかを決めるのが本ファンクションです．コール条件は，5つの事象のうちいくつでも指定でき，そのうちの1つが発生すれば，サブルーチンをコールします．

ソフトウェアドライバーは次の手順でサブルーチンをコールします．マウスから割込みが発生すると，制御がソフトウェアドライバに移ります．ソフトウェアドライバは，コール条件が満たされているかどうかを調べます．満足されていない場合には，そのまま次の動作に移りますが，満足されている場合にはCALL FAR-PROC命令によってサブルーチンに制御を移します．したがってサブルーチンからソフトウェアドライバに制御を戻すためには，RET FAR-PROC命令を使用しなければなりません．

また，サブルーチンをコールする時には，各レジスタには次の情報を格納しています．

AXレジスタ＝コールの原因となった現象

＝1：カーソルの位置が変わった

＝2：左ボタンが押された

＝4：左ボタンが離された

＝8：右ボタンが押された

=16：右ボタンが離された

BLレジスタ=左ボタンの状態

=0　：離されている

=−1：押されている

BHレジスタ=右ボタンの状態

=0　：離されている

=−1：押されている

CXレジスタ=カーソルの位置の水平座標

DXレジスタ=カーソルの位置の垂直座標

○ファンクション15（ミッキー/ドット比の設定）

〔入　力〕

AX=15（機能コード）

CX=水平方向のミッキー/ドット比

DX=垂直方向のミッキー/ドット比

〔出　力〕

なし

〔説　明〕

マウスの移動距離とそれに対応するカーソルの移動距離との比を設定するためのファンクションです．水平方向及び垂直方向にカーソルが8ドット移動するのに要するマウスの水平方向及び垂直方向の移動距離（ミッキー/ドット比）をミッキーの単位で設定します．これにより，マウスを少し動かしただけでカーソルが大きく移動したり，逆にマウスをかなり動かしてもカーソルは少ししか移動しないというように，マウスの感度を変えることができます．

○ファンクション16（水平方向のカーソル移動範囲の設定）

〔入　力〕

AX=16（機能コード）

CX=カーソルの水平方向の移動範囲の最小値（0～639）

DX=カーソルの水平方向の移動範囲の最大値（0～639）

〔出　力〕

なし

〔説　明〕

カーソルの水平方向の移動範囲を設定するためのファンクションです．移動範囲は，その最小値及び最大値を設定することにより決まり，カーソルの中心点がこの範囲内を移動することができます．CXレジスタの値の方がDXレジスタの値より大きい場合には，DXレジスタの値が最小値，CXレジスタの値が最大値となります．

○ファンクション17（垂直方向のカーソル移動範囲の設定）

〔入　力〕

AX=17（機能コード）

CX＝カーソルの垂直方向の移動範囲の最小値

0～199：カラーモード

0～399：高分解能カラーモード

DX＝カーソルの垂直方向の移動範囲の最大値

0～199：カラーモード

0～399：高分解能カラーモード

〔出　力〕

なし

〔説　明〕

カーソルの垂直方向の移動範囲を設定するためのファンクションです．移動範囲は，その最小値及び最大値を設定することにより決まり，カーソルの中心点がこの範囲内を移動することができます．CXレジスタの値の方がDXレジスタの値より大きい場合には，DXレジスタの値が最小値，CXレジスタの値が最大値となります．

ファンクション名：水平方向のカーソル移動範囲内の設定，垂直方向のカーソル移動範囲の設定によってカーソルの移動範囲が変更されて，カーソルの位置が移動範囲外になった場合には，ソフトウェアドライバがカーソルを移動範囲内の端に移動します．

○ファンクション18（カーソルの表示画面の設定）

〔入　力〕

AX＝18（機能コード）

BX＝カーソルの表示画面

＝0：プレーン0へ表示

＝1：プレーン1へ表示

＝2：プレーン2へ表示

＝3：プレーン3へ表示

〔出　力〕

なし

〔説　明〕

カーソルの表示画面を設定するためのファンクションです．カーソルの色は表示画面のパレットで設定された色になります．プレーン3が実装されてなく，カーソルの表示画面の設定がプレーン3へ表示の時は，前回の表示画面へカーソルを表示します．

各画面の初期値は

プレーン0：青

プレーン1：赤

プレーン2：緑

プレーン3：灰

注　意

ソフトウェアドライバは，ファンクション1の呼び出しにより次の様にマウス環境を初期設定します．

・カーソル表示：表示しない

・カーソル位置：スクリーンの中心の位置
　高分解能カラーモード：（319，199）
　カラーモード　　　　：（319，99）
・カーソルの形：左上向きの矢印
・カーソルの表示画面：プレーン2
・カーソルの中心点：（0，0）
・カーソル移動範囲：スクリーン全体
　水平方向：0～639
　垂直方向：高分解能カラーモード：0～399
　　　　　　カラーモード：0～199
・ミッキー/ドット比：水平方向，垂直方向ともに8
・マウスの割込み周期：8 ms

16.3 マウス用ソフトウェアドライバの使用方法

「16.2　ファンクション」で説明したマウスドライバの16種類のファンクションの呼び出し方について説明します．

(1) ソフトウェアドライバを呼び出すための準備

ソフトウェアドライバが持つ16種類のファンクションを使用するためには次に示す準備が必要です．

(a) ソフトウェアドライバのロード

まず最初にソフトウェアドライバを機械語領域にロードします．サンプルプログラム（ソフトウェアドライバ呼び出しのための準備）のⒶがこれに該当します．

ソフトウェアドライバは4KBの領域を必要としますので，これを格納するのに十分な機械語領域を確保しておいて下さい．

(b) マウスハードウェア環境の試験とソフトウェアドライバの初期化

次に，マウスハードウェア環境が整っているかの試験を行います．このエントリはソフトウェアドライバの相対&H100番地から存在しています．仮にマウスハードウェア環境に問題がある場合，パラメータにゼロが設定され返されます．

この際，グラフィック画面の解像度をマウス用ソフトウエアドライバに教えます．

　640×400ドットの場合：3
　640×200ドットの場合：0

をパラメータに指定します．

マウスハードウェア環境に問題がない場合，ソフトウェアドライバのファンクションエントリアドレスが割込みベクタテーブルの33_{16}番目のエントリに設定されます．

サンプルプログラム（ソフトウェアドライバ呼び出しのための準備）のⒷがそれに該当します．

(c) ファンクションエントリアドレスの設定

最後にファンクション呼び出しのためのエントリアドレス（機械語領域内相対アドレス）を設定します（上記(b)で設定されたファンクションエントリアドレスを割込みベクタテーブルの

33_{16}番目のエントリから抽出します）．サンプルプログラム（ソフトウェアドライバ呼び出しのための準備）の©がそれに該当します．

これでソフトウェアドライバを呼び出すためのすべての準備が完了しました．

```
10  'load mouse driver.                ┐
20  CLEAR ,&H7F00                      │
30  DEF SEG = &H7F00                   │……………Ⓐ
40  BLOAD "mouse.cod                   │
50  '                                  ┘
60  'check environment.                                              ┐
70  MOUSE.INI = &H100                                                │
80  FLAG% = 3[注]                                                    │………Ⓑ
90  CALL MOUSE.INI(FLAG%)                                            │
100  IF FLAG% = 0 THEN PRINT "BAD ENVIRONMENT." : END                ┘
110  '
120  'set mouse driver offset.
130  DEF SEG = 0                                          ┐
140  INT33=PEEK(&H33* 4) + PEEK(&H33* 4+ 1 )*256           │……………©
150  MOUSE = INT33+ 3                                     │
160  DEF SEG = &H7F00                                     ┘
170  '
```

サンプルプログラム（ソフトウェアドライバ呼び出しのための準備）

（注）グラフィック画面の解像度が640×400ドットの場合３，640×200ドットの場合０を指定します．

(2) ソフトウェアドライバの呼び出し

「16.2　ファンクション」においては，パラメータは次のような名称で説明されています．

AX，BX，CX，DX，ES

実際にドライバを使用する際は，これらのパラメータをそのままAX％，BX％，CX％，DX％，ES％の変数名に書き換えて使用します．

％が付いていることからわかりますが，パラメータに指定する変数名は整数型の変数でなければなりません．

変数名に関しては，ここでは「16.2　ファンクション」のパラメータの説明と対応がとりやすいようにAX％，BX％，CX％，DX％，ES％を使用していますが別の名前を使用しても何ら問題はありません．

なお，パラメータに直接値を指定することはできません．必ず変数を介して指定します．

ソフトウェアドライバの呼び出しは，16種類のすべてのファンクションに対し次の形式でおこないます．

```
180 'call mouse
190 CALL MOUSE (AX%, BX%, CX%, DX%, ES%)
```

ファンクション毎に使用するパラメータは異なりますが，たとえ使用しないパラメータがあってもダミーの変数名を必らず指定しなければなりません．

16.4 サンプルプログラム

このプログラムは次の6種類の機能を持っています．

(a) 画面のクリア（CLS）
(b) 2点間を結ぶ線を引く（LINE）
(c) 2点間を結ぶ箱を描く（BOX）
(d) 2点間を結ぶ円を描く（CIRCLE）
(e) ペイントする（PAINT）
(f) プログラムを終了する

これらの機能は，画面最上段に表示されたCLS，LINE，BOX，CIRCLE，PAINT，EXITの任意の場所にカーソルを位置付け，左あるいは右ボタンを押すことにより選択します．

また各機能を実行する際の色の選択も可能です．これは画面2行目に表示されたBLACK，BLUE，RED，MAGENTA，GREEN，CYAN，YELLOW，WHITEの任意の場所にカーソルを位置付け，左あるいは右ボタンを押すことにより選択します．

たとえば，赤で線を描く場合，まずカーソルをLINEに位置付け左あるいは右ボタンを押し，次にカーソルをREDに位置付け左あるいは右ボタンを押します．

機能を実行する場所も左あるいは右ボタンで指示します．たとえば，赤で線を描く場合は，前述の方法で機能を選択した後，任意の2点で左あるいは右ボタンを押します．

このプログラムは次の4種のファンクションを使用しています．

(a) カーソルの表示（機能コード：1）
(b) カーソルの消去（機能コード：2）
(c) カーソルの取得（機能コード：3）
(d) カーソルの位置の設定（機能コード：4）

なお，このプログラムにおいてはマウス用ソフトウェアドライバをロードするための機械語領域は(7F000)$_{16}$番地からとしています．

実際にこのプログラムを動作させる場合，お手持のシステムのメモリサイズに合わせて調整してください．

```
1000 CLEAR ,&H7F00
1010 MOUSE.INI=&H100
1020 MSEG=&H7F00
1030 GOSUB *SET.SEG
1040 CONSOLE 0,25,0,0
1050 WIDTH 80,25
1060 SCREEN 3,1,0,1
1070 CLS 3
1080 DIM COMSTR$(6),COLSTR$(7)
1090 GOSUB *LOADCOM
1100 WUP=32
1110 FALSE=(1=0)
1120 TRUE=(1=1)
1130 GOSUB *NORMSCRN
1140 IF PEEK(&H100)<>233 THEN GOSUB *LOAD.MOUSE ELSE 1180
1150 FLAG%=3
1160 CALL MOUSE.INI(FLAG%)
1170 IF FLAG%=0 THEN PRINT "MOUSE:Interface card not found or illegal parameter":BEEP:END
1180 ON STOP GOSUB *MEXIT
1190 STOP ON
1200 DEF SEG=0
1210 INT33=PEEK(&H33*4)+PEEK(&H33*4+1)*256
1220 MOUSE=INT33+3
1230 GOSUB *SET.SEG
1240 AX%=0:GOSUB *MOUSE
1250 IF AX%=0 THEN PRINT "MOUSE:Interface card not found":BEEP:END
1260 COMD=2:SWFLAG=FALSE:COL=7:CANF=FALSE
1270 GOSUB *PRCOM
1280 X=320:Y=200:GOSUB *SETXY
1290 GOSUB *SHOWCU
1300 *MLOOP
1310   IF CANF THEN *COMMAND
1320   GOSUB *WAIT.SWITCH
1330 *COMMAND
1340   CANF=FALSE
1350   IF Y>=16 THEN *SKIPCOM
1360     SAVED.COM=COMD
1370     COMD=INT(X/8/8)+1
1380     IF COMD>6 THEN COMD=SAVED.COM:GOTO *MLOOP
1390     GOSUB *PRCOM
1400     ON COMD GOSUB *MCLS,*MLIN1,*MBOX1,*MCIRCLE1,*NUL.PROC,*MEXIT
1410     IF COMD=1 THEN COMD=SAVED.COM:GOSUB *PRCOM
1420   GOTO *MLOOP
1430 *SKIPCOM
1440   SAVED.COL=COL
1450   IF Y>=WUP THEN *SKIPCOL
1460   COL=INT(X/8/8)
1470   IF COL>7 THEN COL=SAVED.COL:GOTO *MLOOP
1480   GOSUB *PRCOM
1490   GOTO *MLOOP
1500 *SKIPCOL
1510   ON COMD GOSUB *MCLS,*MLIN2,*MBOX2,*MCIRCLE2,*MPAINT,*MEXIT
1520   IF CANF THEN *COMMAND
1530   GOTO *MLOOP
1540 *MCLS
1550   GOSUB *HIDECU
1560   SCREEN ,3:CLS 2:GOSUB *NORMSCRN
1570   GOSUB *SHOWCU
1580   RETURN
1590 *MLIN1
1600   GOSUB *WAIT.SWITCH
1610 *MLIN2
1620   X1=X:Y1=Y
1630   IF Y1<WUP THEN CANF=TRUE:RETURN
1640   GOSUB *WAIT.SWITCH
1650   X2=X:Y2=Y
1660   IF Y2<WUP THEN CANF=TRUE:RETURN
1670   GOSUB *HIDECU
1680   LINE(X1,Y1-WUP)-(X2,Y2-WUP),COL
1690   GOSUB *SHOWCU
```

```
1700   CANF=FALSE:RETURN
1710 *MBOX1
1720   GOSUB *WAIT.SWITCH
1730 *MBOX2
1740   X1=X:Y1=Y
1750   IF Y1<WUP THEN CANF=TRUE:RETURN
1760   GOSUB *WAIT.SWITCH
1770   X2=X:Y2=Y
1780   IF Y2<WUP THEN CANF=TRUE:RETURN
1790   GOSUB *HIDECU
1800   LINE(X1,Y1-WUP)-(X2,Y2-WUP),COL,B
1810   GOSUB *SHOWCU
1820   CANF=FALSE:RETURN
1830 *MCIRCLE1
1840   GOSUB *WAIT.SWITCH
1850 *MCIRCLE2
1860   X1=X:Y1=Y
1870   IF Y1<WUP THEN CANF=TRUE:RETURN
1880   GOSUB *WAIT.SWITCH
1890   X2=X:Y2=Y
1900   IF Y2<WUP THEN CANF=TRUE:RETURN
1910   R=SQR((X2-X1)^2+(Y2-Y1)^2)
1920   GOSUB *HIDECU
1930   CIRCLE(X1,Y1-WUP),R,COL
1940   GOSUB *SHOWCU
1950   CANF=FALSE:RETURN
1960 *MPAINT
1970   GOSUB *HIDECU
1980   PAINT(X,Y-WUP),COL
1990   GOSUB *SHOWCU
2000 *NUL.PROC
2010   CANF=FALSE:RETURN
2020 *MEXIT
2030   GOSUB *HIDECU
2040   COLOR 0:CLS:LOCATE 0,0,1
2050   STOP OFF
2060 END
2070 *MOUSE
2080   CALL MOUSE(AX%,BX%,CX%,DX%,ES%)
2090   RETURN
2100 *LOAD.MOUSE
2110   GOSUB *SET.SEG
2120   BLOAD"mouse.cod"
2130   RETURN
2140 *SET.SEG
2150   DEF SEG=MSEG
2160   RETURN
2170 *SHOWCU
2180   AX%=1:GOSUB *MOUSE
2190   RETURN
2200 *HIDECU
2210   AX%=2:GOSUB *MOUSE
2220   RETURN
2230 *SETXY
2240   AX%=4:CX%=X:DX%=Y:GOSUB *MOUSE
2250   RETURN
2260 *GETXY
2270   AX%=3:GOSUB *MOUSE
2280   SW1=AX%:SW2=BX%:X=CX%:Y=DX%
2290   RETURN
2300 *WAIT.SWITCH:GOSUB *GETXY
2310   IF SW1=0 AND SW2=0 THEN SWFLAG=FALSE:GOTO *WAIT.SWITCH
2320   IF SWFLAG THEN GOTO *WAIT.SWITCH
2330   SWFLAG=TRUE
2340   RETURN
2350 *NORMSCRN
2360   SCREEN 3,0
2370   VIEW(0,WUP)-(639,399)
2380   RETURN
2390 *PRCOM
```

```
2400    COLOR 0
2410    FOR I=1 TO 6
2420      IF I=COMD THEN COLOR 4
2430      LOCATE (I-1)*8,0,0
2440      PRINT COMSTR$(I);
2450      COLOR 0
2460    NEXT
2470    FOR I=0 TO 7
2480      IF I=COL THEN COLOR 4
2490      LOCATE I*8,1,0
2500      PRINT COLSTR$(I);
2510      COLOR 0
2520    NEXT
2530    RETURN
2540 *LOADCOM
2550    FOR I=1 TO 6
2560      READ COMSTR$(I)
2570    NEXT
2580    FOR I=0 TO 7
2590      READ COLSTR$(I)
2600    NEXT
2610    RETURN
2620 DATA "CLS","LINE","BOX","CIRCLE","PAINT","EXIT"
2630 DATA "BLACK","BLUE","RED","MAGENTA","GREEN","CYAN","YELLOW","WHITE"
```

第17章

ライトペンの使い方

ライトペン制御関連命令として次の命令が用意されています．

ON PEN GOSUB文……ライトペン割り込み処理ルーチンの開始行を定義します．
PEN ON/OFF/STOP文……ライトペンによる割り込みの許可，禁止，停止を制御します．
PEN関数……ライトペンからの情報を与えます．

この章ではこれらの命令を使用したライトペン制御サンプルプログラムを掲載しています．

このプログラムを実際に動かすことによりライトペンの制御方法あるいはライトペンの使い方を理解してください．なお，各命令の詳細については「BASICリファレンスマニュアル」を参照してください．

```
100 '--- LIGHT PEN sample program ---
110 '
120  DATA triangle.box.box fill.circle.CRT clear.end
130  SCREEN 0.0:COLOR 7.0.0.7
140  CONSOLE 0.25.0.1:WIDTH 80.25:CLS 3
150  DEFINT A-Z
160  XX=500:YY=300:X=500:Y=300
170  DEF FNX=RND*X:DEF FNY=RND*Y
180  DEF FNC=INT(RND*7+1):DEF FNS=RND*100-50
190  DEF FNR=RND*100
200  P$=CHR$(&H87)+CHR$(&H87):P$=P$+P$
210  ON PEN GOSUB *PENSUB
220  PEN ON
230
240  LOCATE 18.0:PRINT "LIGHT PEN demonstration"
250  LOCATE 66.4:PRINT "< MENU >"
260  FOR I=0 TO 5
270    READ MENU$(I)
280    LOCATE 63.I*3+6:PRINT P$
290    LOCATE 63.I*3+7:PRINT P$
300    LOCATE 68.I*3+7:PRINT MENU$(I)
310  NEXT I
320  VIEW(1.10)-(451.198)..7
330  WINDOW(0.0)-(XX.YY)
340  GOTO 340
350 '
360 *PENSUB
370  PX=PEN(1):PY=PEN(2):PEN OFF
380  IF PX<64 OR  PX>67 THEN *EXIT
390  IF  5<PY AND PY<8  THEN GOSUB *TRIANGLE:GOTO *EXIT
400  IF  8<PY AND PY<11 THEN GOSUB *BOX     :GOTO *EXIT
410  IF 11<PY AND PY<14 THEN GOSUB *BOXFILL :GOTO *EXIT
420  IF 14<PY AND PY<17 THEN GOSUB *CIRCL   :GOTO *EXIT
430  IF 17<PY AND PY<20 THEN GOSUB *CRTCLS  :GOTO *EXIT
440  IF 20<PY AND PY<23 THEN GOSUB *ENDER
450 *EXIT:PEN ON :RETURN
460 '
470 *TRIANGLE
480  SX=FNX:SY=FNY:C=FNC
490  LINE (SX.SY)-STEP(FNS*2.FNS).C
```

```
500  LINE -STEP(FNS*2.FNS).C
510  LINE -(SX.SY).C
520  RETURN
530 *BOX
540  LINE (FNX.FNY)-(FNX.FNY).FNC.B
550  RETURN
560 *BOXFILL
570  LINE (FNX.FNY)-STEP(FNS*2.FNS).FNC.BF
580  RETURN
590 *CIRCL
600  CIRCLE (FNX.FNY).FNR.FNC
610  RETURN
620 *CRTCLS
630  CLS 2:RETURN
640 *ENDER
650  PEN OFF:END
```

(1) プログラムの説明

260行から310行でメニューとライトペンでタッチする部分の表示を行っています．ライトペンでタッチする部分は，必ず色が付いていなければなりません．ただし，ライトペンは赤に対する感度が非常に弱いので，赤は使わないようにして下さい．

後はライトペンが押されるたびに処理ルーチン（360行〜450行）に入りペン関数を読み出してどのメニューが選択されたかを判定しています．

なお，このプログラムはカラーモードで作成されています．

(2) このプログラムの使い方

MENU部に表示されている任意のタッチエリア（画面右側の白でぬられた枠）にライトペンを押しつけてください．対応する図形（三角形，箱，ぬりつぶされた箱，円等）が画面に表示されます．

“CRT clear”を選んだ場合，今まで表示されていた図形がクリアされます．

“end”を選ぶとプログラムは自動的に終了します．

第18章

機械語プログラムの作り方

機械語で書かれたプログラムを呼び出す機能として，USR関数とCALL文が用意されています．ここでは，機械語プログラムを作成する際の注意と，USR関数，CALL文との引数の受け渡し方についで説明します．

18.1　機械語プログラム領域の確保

機械語ルーチンをモニタにより作成したり，ファイルからロードする際には，そのためのメモリ領域を確保する必要があります．この機械語プログラム用のメモリ領域はBASICの使用する領域と重複してはなりません．さもないと，BASICの動作が異常になったり，逆にBASICの動作により準備した機械語プログラムが破壊されたりします．メモリ領域の確保はCLEAR文によりBASICの使用するメモリの上限を設定することにより行います．指定された上限以上の領域はBASICは利用しなくなりますので機械語プログラム領域として用いることができます．詳しくは「14.2　機械語プログラムセグメントのメモリ配置」を参照して下さい．

18.2　機械語プログラムの準備

確保した領域に機械語プログラムを準備する主な方法としては次の3つがあります．

(1)　モニタを用いる．

　MONコマンドによりモニタモードに移り，モニタの機能を利用してプログラムを書き込む．

(2)　BLOADによりロードする．

　既にBSAVEによりセーブされている機械語プログラムをロードする．

(3)　POKEによりプログラムを書き込む．

　比較的短いプログラムであれば，DATA文により機械語プログラムをデータの型で表現しておき，それをREAD文で読み出した後POKE文により書き込む．

また機械語プログラムを作成する際には次の点に注意してください．

(1) DEF USR文により実行開始番地を定義する前，POKE文により書き込む前，およびBSAVE，BLOADの前にはDEF SEG文によりセグメントベースを指定する必要があります．DEF USR，POKE，BSAVEならびにBLOADで指定する番地は，直前に実行されたDEF SEG文で指定されたセグメントベースからのオフセット値です．

(2) USR関数またはCALL文により機械語を呼び出す時には，そのプログラムが実行される時のセグメントベースをDEF SEG文で指定する必要があります．USR関数では，この値はDEF USR文を実行した時に指定されていたセグメントベースと同じものでなければなりません．

(3) BASICに制御を戻すためには，プログラムはインタラプトリターン命令（IRET）を用いなければなりません．

(4) 機械語ルーチンに引き渡された引数が文字列であった場合には，引数の値は次のような4バイトからなるストリングディスクリプタと呼ばれる領域のアドレスとなります．

(a) ストリングディスクリプタの最初のバイトはその文字列の長さ（0から255）を保持します．

(b) 2番目のバイトは文字列がどのセグメントに置かれているかを示します（00_Hならシンボルテーブルセグメント，00_H以外ならテキストセグメントです）．テキストセグメントのベースアドレスは600H番地です（DEF SEG＝&H60でベースアドレスの指定ができます）．シンボルテーブルセグメントのベースアドレスは任意の変数をVARPTR関数に引用することにより知ることができます．（DEF SEG＝VARPTR（A，1）でベースアドレスの指定ができます）．

(c) 3番目のバイトは文字列の置かれている先頭アドレスのセグメントベースからのオフセット値の下位8ビットを保持します．

(d) 最後のバイトは文字列の置かれている先頭アドレスのセグメントベースからのオフセット値の上位8ビットを保持します．

機械語ルーチンではこの4バイトのストリングディスクリプタを書き換えてはなりません．またストリングディスクリプタによって指されるストリングの内容は変更することはできますが，その長さを変更してはなりません．

(5) 機械語ルーチンからBASICに戻る時は，呼び出された時とスタックポインタ（SP）の値が等しくなっていなければなりません．

また，各種ベースレジスタおよびBASICインタプリタ・機械語プログラム間インタフェース用レジスタ（BXおよびAL）に関しても同様です．

18.3 USR関数の呼び出し

USR関数は

USR〔〈番号〉〕〔〈引数〉〕

という書式によって呼び出されます．ただし，〈番号〉は0から9までの数値で，〈引数〉は任意の数値式または文字式です．USR関数は書式上〈引数〉を必要としますので，呼び出されたルーチンが引数を必要としない場合でも，ダミーの引数を与える必要があります．

USR関数が呼び出され，対応する機械語ルーチンに制御が移った時，〔AL〕レジスタは，渡された引数の型を示す値を持っています．その値と意味は次のようになっています．

ALの値	引数の型
2	2バイトの整数（2の補数表現）
3	文字列型
4	単精度型
8	倍精度型

また，〔DX〕レジスタにシンボルテーブルセグメントのベースアドレスが入れられて渡されます．このシンボルテーブルセグメントには，変数や次に示すストリングディスクリプタがあります．

引数が文字列の場合には〔BX〕レジスタが浮動小数点アキュムレータ（FAC）と呼ばれる8バイトの領域の5バイト目を指し，そこにストリングディスクリプタの先頭アドレスがセグメントベースとオフセットの形態で入れられています．

引数が数値の場合には〔BX〕レジスタに，浮動小数点アキュムレータ（FAC）の5バイト目または先頭のアドレスが入っています．実際の引数は，このFACの中に各型に応じて以下のように格納されて渡されます．

●整数型のとき

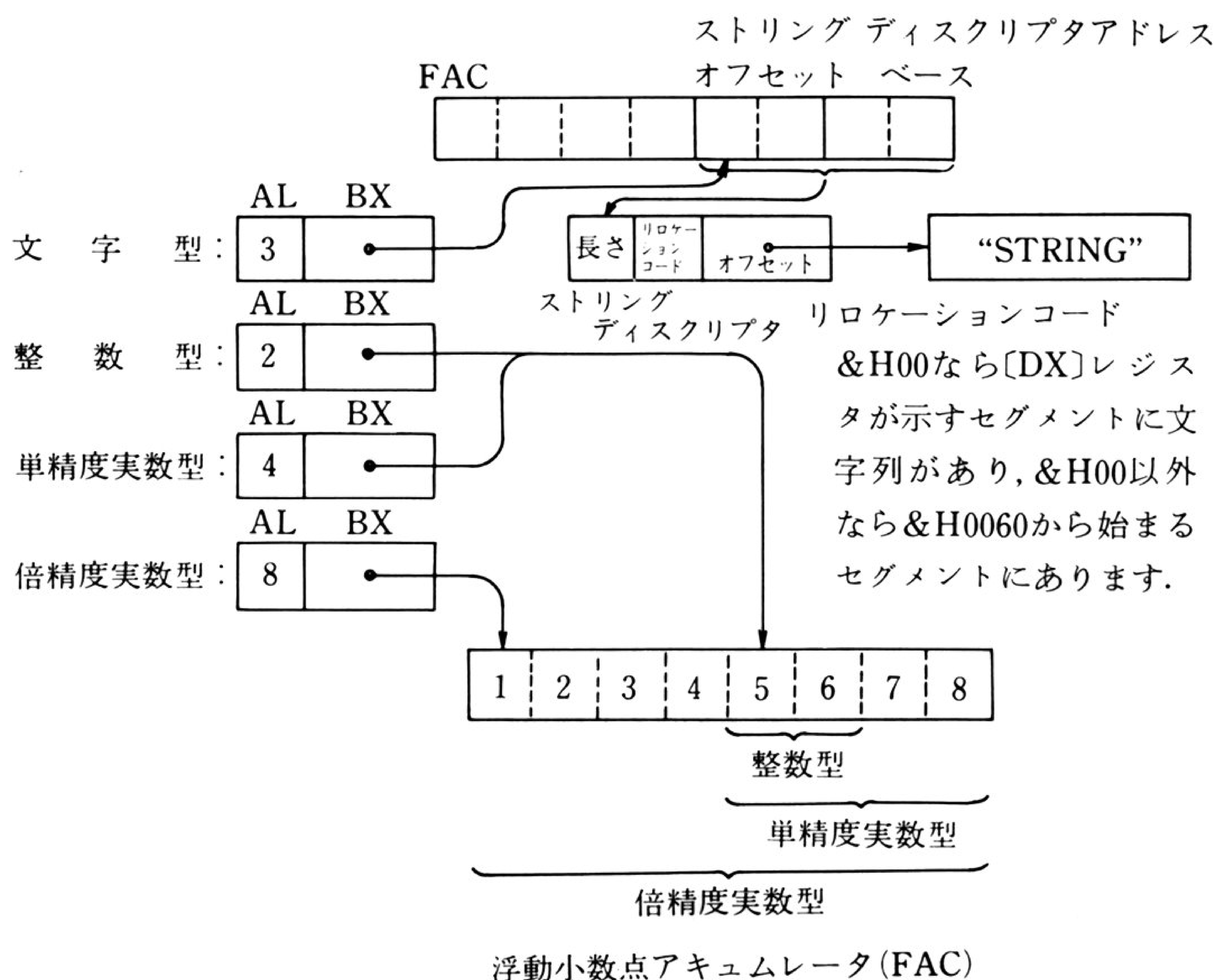
FACの5バイト目を〔BX〕レジスタが指しており，そこに引数の下位8ビットが入り，6バイト目に上位8ビットが入ります．

●単精度実数のとき

FACの5バイト目を〔BX〕レジスタが指しており，5バイト目から8バイト目までの4バイトに内容が入ります．FACの8バイト目は指数部になっており，（指数−128）の値が入ります．小数点は仮数部の最上位ビットの左にあると想定します．7バイト目は仮数部の最高部7ビットを保持します．このバイトの最上位ビットは符号を示し，0で正，1で負を表わします．6バイト目と5バイト目は，それぞれ仮数部の中部と最低部の8ビットを保持します．

●倍精度実数型のとき

FACの1バイト目を〔BX〕レジスタが指しています．5バイト目から8バイト目までは単精度実数型と同じように指数部と仮数部の上位3バイトが入ります．1バイト目から4バイト目には仮数部の下位4バイトが入ります．



浮動小数点アキュムレータ(FAC)

USR関数がBASICに値を返すにはFACに求めた関数値を設定してやることで行えます．このとき〔AL〕レジスタに返す値の型を設定することが必要です．なお，返す値は，引数として渡された値と同じ型でなければなりません．

サンプルプログラム

USR関数の例として，文字列中に含まれる小文字を大文字に変換して返す関数を示します．ここでは，DATA文により機械語プログラムを用意しておき，POKE文で書き込む方法により機械語プログラムを用意しています．

機械語ルーチンでは，まず〔AL〕レジスタの値を調べて，与えられた引数が文字型かどうかを調べています．ここでは，文字型でない場合はそのままBASICに戻ることにしています．

引数が文字型であった場合には，〔BX〕レジスタにより渡されたFACの番地からストリングディスクリプタの番地をもとめ，文字列の長さと，実際に文字列の置かれている番地を求めます．

その後，文字列を最初から1文字ずつ調べていき，小文字であった場合には，D_5ビットを0にすることにより大文字に変換し，文字列を変更します．このように，USR関数は与えられた引数自身を変更するために，USR 0の呼び出し前後でA＄の内容が変化することになります．

```
120 CLEAR,&H7F00
130 MACHINE=&H0
140 DEF SEG=&H7F00
150 DEF USR0=MACHINE
160 '
170 FOR ADR=MACHINE TO MACHINE+49
180   READ BYTE:POKE ADR,BYTE 'poke ML program
190 NEXT ADR
200 '
210 INPUT A$
220   B$=USR0(A$)
230   PRINT B$
240 GOTO 210
250 END
260 '
270 '    machine language program data
280 '
290 DATA &H3C,&H03              :' CMP    AL,03H     ; string ?
300 DATA &H75,&H2D              :' JNZ    STRCNV90   ; jump no
310 DATA &H53                   :' PUSH   BX
320 DATA &H51                   :' PUSH   CX
330 DATA &H52                   :' PUSH   DX
340 DATA &H56                   :' PUSH   SI
345 DATA &HC5,&H1F              :' LDS    BX,[BX]
350 DATA &H8B,&H0F              :' MOV    CX,[BX]    ; CH:rel no.  CL:data length
360 DATA &H80,&HFD,&H00         :' CMP    CH,00H
370 DATA &H74,&H05              :' JC     STRCNV10   ; jump yes
380 DATA &H32,&HED              :' XOR    CH,CH
390 DATA &HBA,&H60,&H00         :' MOV    DX,0060H
400 '                  STRCNV10:'
410 DATA &H8B,&H77,&H02         :' MOV    SI,2[BX]   ; get offset
420 DATA &H8E,&HDA              :' MOV    DS,DX
430 '                  STRCNV20:'
440 DATA &H8A,&H24              :' MOV    AH,[SI]
450 DATA &H80,&HFC,&H61         :' CMP    AH,61H     ; cmp. small "a"
460 DATA &H72,&H08              :' JC     STRCNV30   ; jump less
470 DATA &H80,&HFC,&H7B         :' CMP    AH,7BH     ; cmp. small "z"
480 DATA &H73,&H03              :' JNC    STRCNV30   ; jump greater or equal
490 DATA &H80,&H24,&HDF         :' AND    BYTE PTR [SI],0DFH     ; convert
500 '                  STRCNV30:'
510 DATA &H46                   :' INC    SI
520 DATA &HE2,&HEE              :' LOOP   STRCNV20
530 DATA &H5E                   :' POP    SI
540 DATA &H5A                   :' POP    DX
550 DATA &H59                   :' POP    CX
560 DATA &H5B                   :' POP    BX
570 '                  STRCNV90:'
580 DATA &HCF                   :' IRET
```

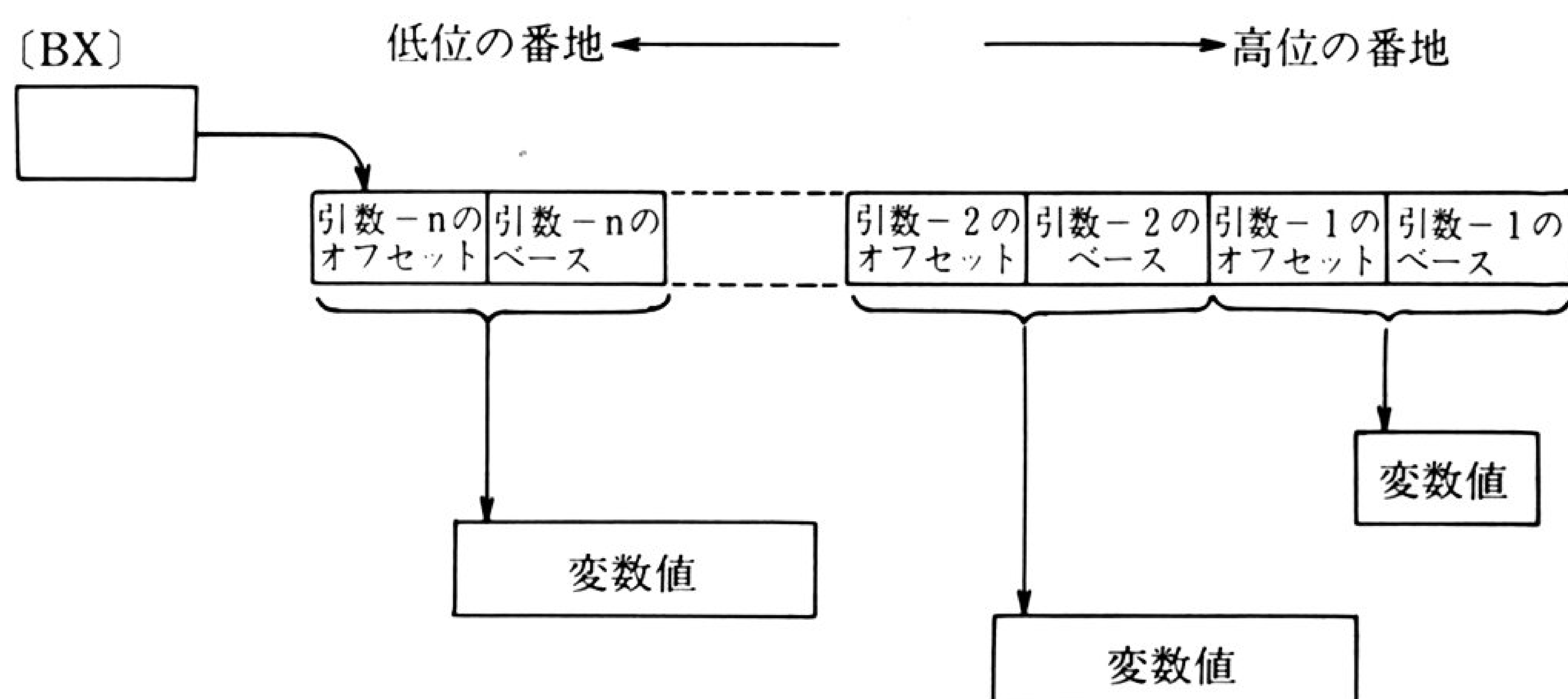
18.4 CALL文

機械語ルーチンはCALL文を用いても呼び出すことができます．

引数を持たないCALL文は機械語のCALL命令と同等であり，指定された番地からのルーチンが呼び出されます．BASICに戻るには，USR関数と同じようにIRET命令で戻ります．

CALL文が引数を持っている場合には，〔DS〕レジスタおよび〔BX〕レジスタにそれぞれ引数で指定された変数の置かれている番地のテーブルの入っている領域のベースアドレスとテーブルの開始番地のオフセット値が準備された上で機械語が呼び出されます．

このテーブルの形式は次のようになっています．



〔DS〕：引数テーブルのベースアドレス

引数テーブルは，引数の値の入っている領域の開始番地のセグメントベースアドレスとオフセット値が組になっており，高位の番地から低位の番地に向けて順次格納されています．

CALL文を用いる際には，次の事項に注意してください．

- CALL文が渡す引数の番地は，USR関数の場合のFACの番地ではなく，VARPTR関数の値と同じ変数値の置かれている番地です．
- CALL文では複数の引数を与えることができます．また，与えられた引数の番地を介してBASICに値を返すことができます．
- 引数が文字型の場合には準備される番地はストリングディスクリプタの番地です．
- CALL文は引数の個数や型については何ら情報を与えません．従って，これらを一致させることに注意しなければなりません．

サンプルプログラム

CALL文の例として，2つの数の最大値を与えるルーチンを示します．A％，B％により渡された2つの数のうち大きい方の値をC％に代入します．

機械語ルーチンでは〔BX〕レジスタの示す引数テーブルからA％，B％の内容を取り出し，大きな方をC％の番地に格納します．

```
10 '   CALL statement sample program
20 '
30 CLEAR ,&H7D00
40 DEF SEG=&H7D00
50 MAX=&H0
60 '
70 FOR ADR=MAX TO MAX+47
80   READ BYTE:POKE ADR,BYTE  'poke ML language
90 NEXT ADR
100 '
110 C%=0
120 INPUT A%,B%
130   CALL MAX(A%,B%,C%)
140   PRINT C%
150 GOTO 120
160 END
170 '
180 '  machine language program data
190 '
200 '                    MAX:
210 DATA &H50               :' PUSH  AX
220 DATA &H53               :' PUSH  BX
230 DATA &H51               :' PUSH  CX
240 DATA &H56               :' PUSH  SI
250 DATA &H06               :' PUSH  ES
260 DATA &H8B,&H4F,&H0A     :' MOV   CX,10[BX]     ; get 1st base addr.
270 DATA &H8E,&HC1          :' MOV   ES,CX
280 DATA &H8B,&H77,&H08     :' MOV   SI,8[BX]      ; get 1st offset addr.
290 DATA &H26,&H8B,&H04     :' MOV   AX,ES:[SI]    ; get 1st data
300 DATA &H8B,&H4F,&H06     :' MOV   CX,6[BX]      ; get 2nd base addr.
310 DATA &H8E,&HC1          :' MOV   ES,CX
320 DATA &H8B,&H77,&H04     :' MOV   SI,4[BX]      ; get 2nd offset addr.
330 DATA &H26,&H3B,&H04     :' CMP   AX,ES:[SI]    ; compare data
340 DATA &H7D,&H03          :' JGE   CMP10         ; jump 1st >=2nd
350 DATA &H26,&H8B,&H04     :' MOV   AX,ES:[SI]    ; move 2nd to AX
360 '                  CMP10:'
370 DATA &H8B,&H4F,&H02     :' MOV   CX,2[BX]      ; get 3rd base addr.
380 DATA &H8E,&HC1          :' MOV   ES,CX
390 DATA &H8B,&H37          :' MOV   SI,0[BX]      ; get 3rd offset addr.
400 DATA &H26,&H89,&H04     :' MOV   ES:[SI],AX    ; move AX to 3rd area
410 DATA &H07               :' POP   ES
420 DATA &H5E               :' POP   SI
430 DATA &H59               :' POP   CX
440 DATA &H5B               :' POP   BX
450 DATA &H58               :' POP   AX
460 DATA &HCF               :' IRET

run
? 100,200
 200
? 100,-200
 100
?
^C
Break in 120
Ok
```

第19章

拡張機能

N_{88}-BASIC(86) の拡張機能として次の4つの機能があります.

項番	拡張機能	必要な機器	使用宣言の方法	システムコードによる利用者領域の圧迫
1	モデム-NCU内蔵電話機制御機能	PC-TL101, PC-TL102, PC-9863, PC-9865, DATAX ITM1200, ITM1212のいずれか	メモリスイッチ SW6 2^5ビット：ON	利用者領域が20KB減少します
2	サウンド機能	本体標準装備	メモリスイッチ SW4 2^3ビット：ON (標準：ON)	プログラム領域が2KB減少します
3	GP-IB (IEEE-488) インタフェース制御機能	PC-9801-29N GP-IB (IEEE-488) インタフェースボード	メモリスイッチ SW4 2^5ビット：ON	なし
4	RS-232Cインタフェース（2回線/3回線）制御機能	PC-9861K RS-232C（2回線/3回線）インタフェースボード	メモリスイッチ SW4 2^4ビット：ON	なし

備考 メモリスイッチの設定については，システムディスクに添付されているユーティリティ「switch.n88」を御使用下さい.

これらの拡張機能のうち，モデム-NCU内蔵電話機制御機能とサウンド機能については標準装備となっております.

GP-IB (IEEE-488) インタフェース制御機能およびRS-232Cインタフェース（2回線/3回線）制御機能については別売のインタフェースボードを実装することにより使用可能となります.

この章では標準装備のモデム-NCU内蔵電話機制御機能およびサウンド機能について詳細に解説します.

GP-IB (IEEE-488) インタフェース制御機能およびRS-232Cインタフェース（2回線/3回線）制御機能については各インタフェースボードに添付されている取扱い説明書を御参照下さい.

19.1 モデム-NCU内蔵電話機制御機能

N₈₈-日本語BASIC(86)はモデム-NCU内蔵電話PC-TL101, PC-TL102オートホン, PC-9863, PC-9865モデムボード＋PC-TL901ハンドセットあるいはインテリジェントモデムDATAX ITM1200, ITM1212等の制御機能を持っており，オートダイアル，自動着信さらにはモデムを利用したデータ通信等を簡単に実現することができます.

また，N₈₈-日本語BASIC(86) システムディスクには，電話帳の作成や保守，電話の自動発信，BBS接続，ファイル転送機能等を持つ電話管理ユーティリティ(tele.n88) が格納されておりますので御利用下さい.

ここでは，電話制御のために拡張された拡張電話制御命令の機能および使用方法について解説しています.

また，電話管理ユーティリティ (tele.n88) の使用方法についても解説しています.

19.1.1 拡張電話制御命令の概要

N₈₈-日本語BASIC(86) は，PC-TL101, PC-TL102オートホン，PC-9863, PC-9865モデムボード＋PC-TL901ハンドセットあるいはDATAX ITM1200, ITM1212等を使用したオートダイアル，自動着信，種々のデータ通信を実現するため次の示す拡張命令（コマンドおよび関数）を提供します.

・電話機の制御命令

CMD LINE OPEN（電話機とBASICを論理的に接続する）
CMD LINE CLOSE（電話機とBASICを論理的に切り離す）
CMD DIAL（電話をかける）
CMD STORE DIAL（電話機に電話番号を記憶させる）[注]
CMD ON LINE GOSUB（着信があった時の割り込み先を定義する）
CMD LINE ON/OFF/STOP（着信割り込みを制御する）
CMD RECEIVE（着信があった場合の動作を指定する）
CMD MODE CUT（電話を切る）

・電話機に対する関数

STATUS LINE（着信があったかどうかを調べる）
STATUS DIAL$（電話機に記憶されている電話番号を調べる）[注]
STATUS DIAL（電話機に記憶されている電話番号の機能を調べる）[注]
STATUS MODE（現在の電話機のモードを調べる）

・通信関係の命令

CMD CHANGE DUPLEX（通信方式を切り換える）
CMD BREAK（ブレーク信号を送出する）
CMD ERROR ON/OFF/STOP（通信エラー割り込みを制御する）
CMD ON ERROR GOSUB（通信エラーがあった時の割り込み先を定義する）

・通信関係の関数

STATUS ERROR（通信エラーの有無あるいはエラーの種類を調べる）

注：これらの命令あるいは関数はPC-9863, PC-9865モデムボードにおいては使用できません.

これらの命令を使用することにより，オートダイアル制御，自動着信制御および種々のデータ通信が簡単に実現できます．

これらの命令の機能あるいは具体的な使用方法については「19.1.4　拡張電話制御命令の使用例」および「19.1.5　拡張電話制御命令」を参照してください．

19.1.2　拡張電話制御命令の使用に際して

(1)　電話機のスイッチ設定

PC-TL101, PC-TL102オートホン，PC-9863, PC-9865モデムボード＋PC-TL901ハンドセットあるいはDATAX ITM1200, ITM1212のユーザーズマニュアルを参照しながら，転送速度，発進/受信モード等を設定してください．

(2)　電話機と本体の接続

電話機と本体（RS-232C標準ポート（第1回線）および拡張ポート（第2回線/第3回線……この場合，RS-232C拡張インタフェースボード（PC-9861K）が必要です）のいずれも使用できます）を電話機付属のRS-232Cケーブルで確実に接続してください．

次に，PC-TL101, PC-TL102オートホンあるいはDATAX ITM1200, ITM1212の場合，電話機の自動/手動ボタンを押して“自動”のランプが点灯するようにセットしてください（“手動”の場合，オートダイアル自動着信等の制御はおこなえません）．

(3)　拡張電話制御命令の使用宣言

メモリスイッチSW6の2^5ビットをONにします．

このスイッチをONにしておかないと，拡張電話制御命令は使用できません（"Feature not available"エラーとなります）．

このスイッチの設定はシステムディスクに格納されているユーティリティswitch.n88により簡単に行えます．

① まず，本体のディップSW2の5番スイッチを下にたおした状態でN_{88}-日本語BASIC(86)を立上げます．

② “menu”プログラムを起動し，“メモリスイッチの設定”（switch.n88）を選択します．

③ switch.n88が起動されたら，“Basic mode”を選択します．

```
……Basic mode……
  電話制御命令の使用……………………………………………………
                    （使わない）        使う
  拡張モニタモード………………………………………………………
                    （使わない）        使う
```

電話制御命令の使用を“使う”にします．（[→]キーでカーサを“使う”に移動し，[↵]キーで決定します）

④ ESC キーを押し，初期画面に戻ります．

⑤ システムを再起動（リセット）します．拡張電話制御命令が使用できます．

注意 拡張電話制御命令を使用する場合，利用者メモリは最低256KB必要です（拡張電話制御命令の使用を宣言しますと，使用しない場合にくらベシステムコードが約20KB増大します）．

なお，ここで「(4) 転送速度の指定」で解説しているメモリスイッチSW2に対する転送速度の指定を行ってもかまいません．

(4) 転送速度の指定

① RS-232C標準ポート（第1回線）使用の場合

第1回線を使用する場合，電話機の転送速度と同じ値をメモリスイッチSW2の2^0～2^3ビットに設定します（300ボーあるいは1200ボーのいずれかを指定します）

このスイッチの設定は拡張電話制御命令の使用宣言の場合と同様にswitch.n88ユーティリティで簡単に行えます．

(a) 拡張電話制御命令の使用宣言の場合と同様の方法でswitch.n88ユーティリティを起動し，“RS-232C（初期設定）”を選択します．

```
……RS-232C（初期設定）……
 Xパラメータ………………………………………………………
                    （無効）          有効
 通信方法……………………………………………………………
                    （全二重）        半二重
 データbit長 …………………………………………………………
                    （7bit）          8bit
 パリティ　チェック………………………………………………
                    （使わない）  奇数    偶数
 ストップbit長 ………………………………………………………
                    （1bit）     1.5bit    2bit
 ボーレート……………………………………………………………
                    75       150     300     600
                    （1200）  2400    4800    9600
```

300あるいは1200を選択します（→ キーあるいは ← キーでカーサを“300”あるいは“1200”に位置付け，↵ キーで決定します）．

(b) ESC キーを押し初期画面に戻ります．

(c) システムを再起動（リセット）します．

② RS-232C第2回線あるいは第3回線使用の場合

第2回線あるいは第3回線を使用する場合，RS-232C拡張インタフェースボード（PC-9861K）上の転送速度設定スイッチの値を電話機の転送速度と同じ値（300ボーあるいは1200ボー）に設定します．

19.1.3　4つのモード

(1)　初期モード

BASICとモデム内蔵電話機が論理的に接続されていない状態です．この状態ではオートダイアル，自動着信等の機能は利用できません．手動による電話回線の接続あるいはモデム内蔵電話機のモデム機能のみを利用したデータ通信が可能です．

CMD LINE OPEN命令により電話機とBASICが論理的に接続され，オートダイアル，自動着信等の機能が利用できるオフラインコマンドモードとなります．

〔初期モードで使用可能な拡張電話制御命令〕

CMD LINE OPEN
STATUS MODE
コミュニケーションファイルに対する入出力命令（OPEN/CLOSE/PRINT #/INPUT #等）
CMD CHANGE DUPLEX

(2)　オフラインコマンドモード

モデム内蔵電話機と電話回線は切り離されていますがBASICとモデム内蔵電話機との間でコマンドの受け渡しができるモードです．

〔オフラインコマンドモードで使用可能な拡張電話制御命令〕

CMD LINE CLOSE
CMD STORE DIAL[注]
CMD DIAL
CMD RECEIVE
CMD LINE ON/OFF/STOP
CMD MODE CUT
CMD ON LINE GOSUB
STATUS LINE
STATUS DIAL$[注]
STATUS DIAL[注]
STATUS MODE

注：これらの命令あるいは関数はPC-9863, PC-9865モデムボードにおいては使用できません．

(3) データ通信モード

電話回線とBASICがモデムを介して接続された状態であり，コミュニケーションファイルを通じて電話回線上の通信相手とデータの受け渡しができるモードです．

この状態では電話機のフックランプとデータランプがともに点灯しています．

またこの状態で受話器を外し，通話/データ切り替えボタンを押すことにより通話モードに移行することができます．

〔データ通信モードで使用可能な拡張電話制御命令〕

コミュニケーションファイルに対する入出力命令（OPEN/CLOSE/PRINT#/INPUT#他）

CMD CHANGE DUPLEX

CMD BREAK

CMD MODE CUT

CMD LINE CLOSE

STATUS ERROR

STATUS MODE

(4) 通話モード

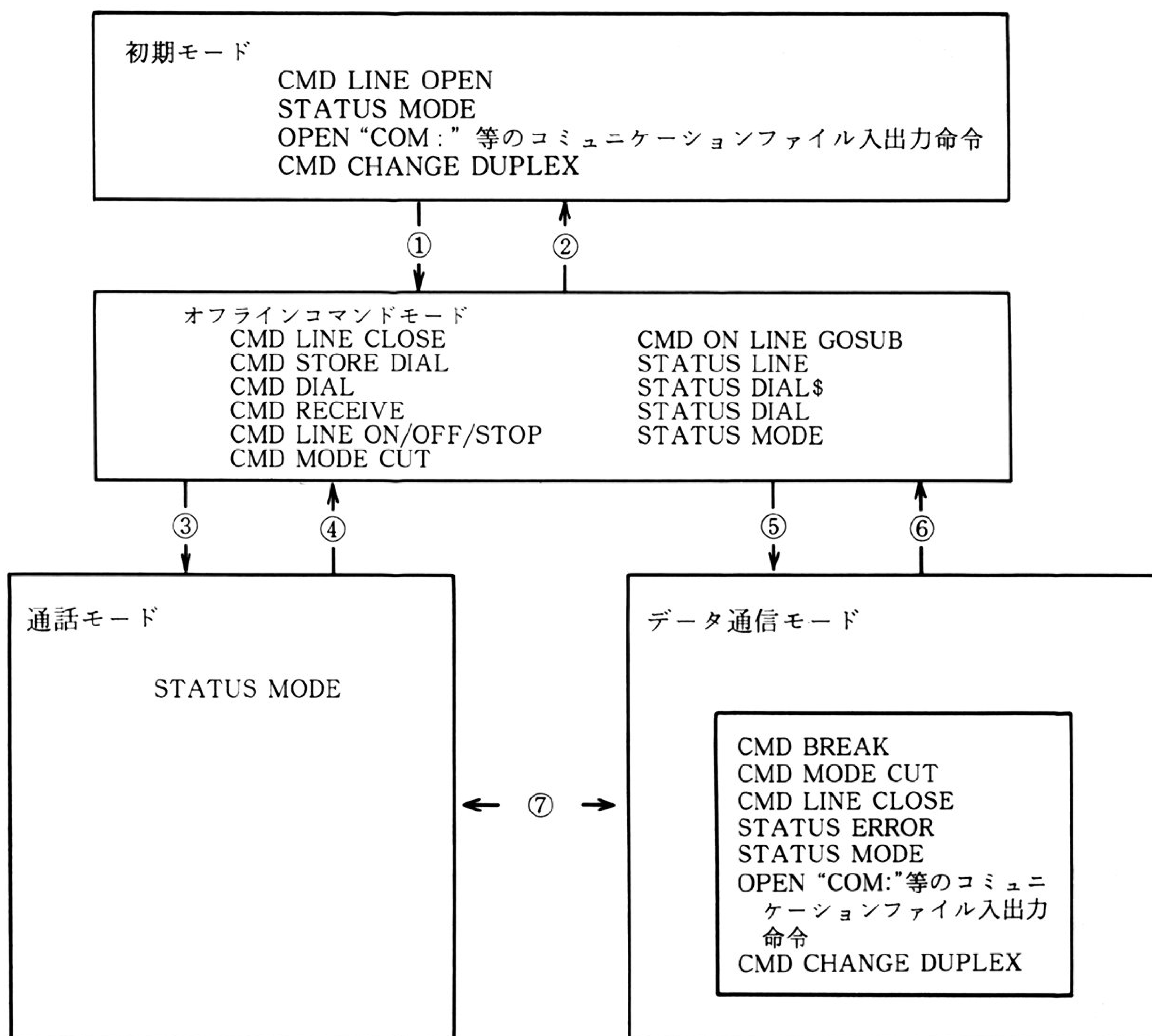
電話回線とモデム内蔵電話機とが接続され，通話相手と話すことができるモードです．フックランプが点灯し，データランプは消灯しています．

この状態では，BASICから電話機を制御することはできません．受話器を置くかフックボタンを押すことによって電話を切るとオフラインコマンドモードに戻ります．

またこのモードで通話/データ切り替えボタンを押すことによりデータ通信モードに変わります．ただしボタンを押す前にCMD LINE OPEN命令が実行されていなければなりません．

〔通話モードで使用可能な拡張電話制御命令〕

STATUS MODE



各モードの遷移図

①：CMD LINE OPENによる

②：CMD LINE CLOSEによる

③：CMD DIAL，手動ダイアルまたは手動着信による

④：受話器を置くまたはフックボタンを押す

⑤：CMD DIALまたは自動着信による

⑥：CMD MODE CUTによる

⑦：通話/データ切り換えボタンによる

19.1.4 拡張電話制御命令の使用例

(1) 電話をかけて通話を行う

以下に自動ダイアルを行うプログラムの例を示します．

```
100 ' AUTO DIAL
110 CMD LINE OPEN "COM:"
120 *MAIN
130 INPUT "Telephone number";DL$
140 IF DL$="" THEN CMD LINE CLOSE:END
150 CMD DIAL DL$
160 GOTO *MAIN
```

それでは各命令について説明します

110 CMD LINE OPEN "COM:"

これは電話機とBASICを論理的に接続し，初期モードからオフラインコマンドモードに切り替える命令です．"COM:" は，本体と電話機がRS-232C標準ポートで接続されていることを意味します．

130 INPUT "Telephone number "; DL$

140 IF DL$="" THEN CMD LINE CLOSE : END

ここでは相手の電話番号を入力します．(内線電話から外線電話をかける場合に " 0 " などを先頭に指定する必要がある場合には "0：×××" のように指定してください)

電話番号を省略した場合には電話機とBASICとを切り離し，初期状態に戻してプログラムを終了します．

150 CMD DIAL DL$

これは電話をかける命令です．この場合ダイアル後は次の命令に進み，再び電話番号入力待ちとなります．

オートダイアルをおこないますと，電話機のフックランプが自動的に点燈し，相手を呼び出し始めます．この間，内蔵スピーカを通して呼び出し音が確認できます．相手が出たら，受話器をとって会話します．通話が終わった場合，または話し中あるいは相手が出ない等の場合には受話器を置くかフックボタンを押して電話を切ってください．

(2)　電話をかけてデータを送信する

以下に自動ダイアルを行い，ファイルの内容を自動転送するプログラムの例を示します．

```
100 ' AUTO DIAL & SEND FILE
110 CMD LINE OPEN "COM:"
120 *MAIN
130 INPUT "Telephone number";DL$
140 IF DL$="" THEN CMD LINE CLOSE:END
150 INPUT "Send file name";FLN$
160 OPEN FLN$ FOR INPUT AS #1
170 CMD DIAL DL$,1
180 OPEN "COM:E71N" AS #2
190 PRINT #2,FLN$
200 WHILE (NOT EOF(1))
210  PRINT #2,"1"
220  LINE INPUT #1,A$
230  PRINT #2,A$
240 WEND
250 PRINT #2,"0"
260 CLOSE
270 CMD MODE CUT
280 GOTO *MAIN
```

それでは各命令について説明します．

110　CMD LINE OPEN "COM:"

これは電話機とBASICを論理的に接続し，初期状態からオフラインコマンドモードにする命令です．

130　INPUT "Telephone number ";DL$

140　IF DL$="" THEN CMD LINE CLOSE:END

ここでは相手の電話番号を入力します．

電話番号を省略した場合には電話機とBASICとを切り離し，初期状態に戻してプログラムを終了します．

150　INPUT "Send file name ";FLN$

160　OPEN FLN$ FOR INPUT AS ♯1

次に転送するファイル名を入力しそのファイルをオープンします．

170　CMD DIAL DL$, 1

電話をかける命令です．この場合“1”が指定されていますのでダイアル後相手のアンサートーンを検出するまで待っています．

この命令を実行するためには送信先が(3)の受信プログラムを起動して受信待ちの状態になっていなければなりません．そうでない場合には規定時間内にアンサートーンが検出できないのでエラーとなります．詳しくはCMD DIALコマンドを参照してください．

180　OPEN "COM:E71N" AS ♯2

電話機に接続されているコミュニケーションファイルをオープンします．

この命令を実行することにより以降，電話機内蔵のモデムを介してPRINT♯，INPUT♯等の入出力命令を実行することが可能となります．

なお，PC-TL101，PC-9863あるいはITM1200を使用し1200ボーによる転送をおこなう場合，OPEN命令の前に次の命令が必要になります．

CMD CHANGE DUPLEX 1, 1

この命令は通信方式を全二重通信方式から半二重通信方式に切り換える命令です．

これらの機器では，1200ボーの通信は半二重方式で制御されます．

```
190  PRINT  #2,FLN$
```

先ず相手に，転送するファイル名を伝えます．

```
200  WHILE (NOT EOF(1))
210    PRINT  #2,"1"
220    LINE INPUT  #1,A$
230    PRINT  #2,A$
240  WEND
250  PRINT  #2,"0"
```

ここでファイルのデータを順次相手に送信します．このプログラムでは“1”のレコードのあとにはファイルの内容を1レコード分送信し“0”のレコードのあとにはデータがないという約束を決めています．

```
260  CLOSE
```

電話機に接続されているコミュニケーションファイルと転送したデータファイルをクローズします．

```
270  CMD MODE CUT
```

電話を切りオフラインコマンドモードに移行させる命令です．

(3)　自動着信後データを受信する

自動着信を行い，受信したデータをファイルに書き出すプログラムの例を示します．このプログラムは(2)の例題プログラムで電話をかけることを想定しています．

```
100 ' RECEIVE FILE
110 CMD LINE OPEN "COM:"
120 CMD ON LINE GOSUB *REC
130 CMD LINE ON
140 *LOOP
150 GOTO *LOOP
160 *REC
170 CMD RECEIVE 1
180 IF STATUS MODE<>2 THEN 180
190 OPEN "COM:E71N" AS #1
200 LINE INPUT #1,FLN$
210 PRINT "RECEIVE FILE=";FLN$
220 PRINT "START"
230 OPEN FLN$ FOR OUTPUT AS #2
240 *REC10
250 INPUT #1,A$
260 IF A$<>"1" THEN *REC20
270 LINE INPUT #1,A$
280 PRINT #2,A$
290 GOTO *REC10
300 *REC20
310 CLOSE
320 CMD MODE CUT
330 CMD RECEIVE 0
340 PRINT "END"
350 RETURN
```

それでは各命令について説明します．

110　CMD LINE OPEN "COM:"

これは電話機とBASICを論理的に接続し，初期状態からオフラインコマンドモードにする命令です．

120　CMD ON LINE GOSUB　*REC

130　CMD LINE ON

ここでは着信があった場合の分岐先を定義し，着信割り込みを許可します．

これ以降着信があるとすぐ*RECルーチンへ分岐します．

140　*LOOP

150　GOTO　*LOOP

ここは着信があるまで待っています．もちろんこの間に別の処理を実行させることも可能です．

160　*REC

170　CMD RECEIVE 1

180　IF STATUS MODE<>2 THEN 180

190　OPEN "COM:E71N" AS　#1

着信があった場合の処理ルーチンです．

ここでは，まず自動着信に切り換えます．現在着信割り込みを検出していますからデータ通信モードに移行しますが，アンサートーンを返し終わるまでの間はオフラインコマンドモードの状態ですので確実にデータ通信モードになるまで待っています．

次に電話機に接続されているコミュニケーションファイルをオープンします．以降電話機に内

蔵されたモデムを介してPRINT#，INPUT#等の入出力命令が実行可能となります．なお，PC-TL101, PC-9863あるいはITM1200を使用し，1200ボーによる通信をおこなう場合，OPEN命令の前に次の命令が必要になります．

```
CMD CHANGE DUPLEX 1, 1
```

この命令は，通信方式を全二重方式から半二重方式に切り換える命令です．

これらの機器では，1200ボーの通信は半二重方式で制御されます．

```
200 LINE INPUT #1,FLN$
210 PRINT "RECEIVE FILE=";FLN$
220 PRINT "START"
230 OPEN FLN$ FOR OUTPUT AS #2
```

まずコミュニケーションファイルから，受信するファイルの名前を受け取りそのファイルを作成（オープン）します．同時に画面上にその旨表示します．

```
240 *REC10
250 INPUT #1,A$
260 IF A$<>"1" THEN *REC20
270 LINE INPUT #1,A$
280 PRINT #2,A$
290 GOTO *REC10
```

コミュニケーションファイルから順次データを受信し，ファイルに書き込んでいきます．

このとき，先の(2)の例題プログラムで説明したように"１"のレコードの次にはファイルの内容が送られてきますので次のレコードをファイルに書き込みます．"１"以外のレコードを受信した場合には転送完了です．

```
310 CLOSE
```

ここで使用したファイルを全てクローズします．

```
320 CMD MODE CUT
330 CMD RECEIVE 0
```

電話を切って次の着信を待ちます．

ここで自動着信を禁止しておきます．つまり，次の着信割り込み処理ルーチンに分岐するまではデータ通信モードに切り換えないように処理しています．

```
340 PRINT "END"
350 RETURN
```

受信終了のメッセージを表示して割り込み前の処理（着信待ち）に戻します．

19.1.5　拡張電話制御命令

ここでは，各拡張電話制御命令についてアルファベット順に説明しています．

項番	コマンド名あるいは関数名	機　　　能	コマンド/関数
1	CMD BREAK	ブレーク信号を送出する	コマンド
2	CMD CHANGE DUPLEX	通信方式を切り換える	コマンド
3	CMD DIAL	電話をかける	コマンド
4	CMD ERROR ON/OFF/STOP	通信エラー割り込みを制御する	コマンド
5	CMD LINE CLOSE	電話機とBASICを論理的に切り離す	コマンド
6	CMD LINE ON/OFF/STOP	着信割り込みを制御する	コマンド
7	CMD LINE OPEN	電話機とBASICを論理的に接続する	コマンド
8	CMD MODE CUT	電話を切る	コマンド
9	CMD ON ERROR GOSUB	通信エラー時の割り込み先を定義する	コマンド
10	CMD ON LINE GOSUB	着信があった時の割り込み先を定義する	コマンド
11	CMD RECEIVE	着信があった場合の動作を指定する	コマンド
12	CMD STORE DIAL[注]	電話機に電話番号を記憶させる	コマンド
13	STATUS DIAL[注]	電話機に記憶されている電話番号の機能を調べる	関数
14	STATUS DIAL$[注]	電話機に記憶されている電話番号を調べる	関数
15	STATUS ERROR	通信エラーの有無/種類を調べる	関数
16	STATUS LINE	着信の有無を調べる	関数
17	STATUS MODE	現在の電話機のモードを調べる	関数

注：これらの命令あるいは関数はPC-9863, PC-9865モデムボードにおいては使用できません．

各命令の説明は次の構成で行われています．

命令の名前
- 機　能
- 書　式
- 文　例
- 解　説
- サンプルプログラム

機　能　命令の機能を簡単に示します．

書　式　命令の記述の仕方を示します．実際の入力時には次のような決まりに従ってください．

1．アルファベットの大文字で示された項目は，そのまま入力します．入力する場合は，小文字でも大文字でもかまいません．ただし，ダブルクォーテーション（”）で囲まれた文字例（ファイル名以外）は，大文字と小文字を区別してください．

2．カギカッコ“＜”，“＞”で囲まれた項目は，ユーザーが指定します．

3．角カッコ“〔”,“〕”で囲まれた項目は，省略することができます．省略した場合，デフォルト値（BASICであらかじめ設定されている値）が適用されます．
拡張電話制御命令で省略できるパラメータは〈電話機番号〉のみです．〈電話機番号〉が省略された場合，番号は1となります．

4．上記のカギカッコ，角カッコ以外の記号でカッコ（　），コンマ，シャープ記号（＃）などの記号は示された位置に正しく入力します．

文　例　実際の入力の仕方の見本として簡単な例を示します．

解　説　命令の使用方法や詳しい機能とそれに関して注意しなければならない点などを説明します．

サンプルプログラム　いくつかの文を交えたプログラム例を示します．

CMD BREAK

機　能　ブレイク信号の送出を制御します．

書　式　1）CMD BREAK [[#]〈電話機番号〉,]〈時間〉

2）CMD BREAK [[#]〈電話機番号〉] ON

3）CMD BREAK [[#]〈電話機番号〉] OFF

文　例　CMD BREAK　#1,2

解　説

1）　指定された〈時間〉だけブレイク信号を電話機に発信します．〈時間〉の単位は１秒単位で１～10の値が指定できます．

2）ブレイク信号の発信を開始します．

この命令を実行するとCMD BREAK OFFを実行するまでブレイク信号を電話機に送出し続けます．

3）ブレイク信号の発信を終了します．

CMD BREAK ONにより送出されていたブレイク信号を止めます．

この命令はデータ通信モードでのみ有効です．

サンプルプログラム

```
100 CMD LINE OPEN "COM1:" AS #1
110 INPUT "電話番号",TELNO$
120 CMD DIAL #1,TELNO$,2
130 OPEN "COM1:E71N" AS #1
140 CMD BREAK #1,3
150 ON STOP GOSUB *ISTP:STOP ON
160 OPEN "KYBD:" FOR INPUT AS #2
170 IF LOC(1)<>0 THEN PRINT INPUT$(1,1);
180 IF LOC(2)<>0 THEN A$=INPUT$(1,2):PRINT A$;:PRINT #1,A$;
190 GOTO 170
200 *ISTP
210 CMD MODE CUT #1
220 CMD LINE CLOSE #1
230 END
```

CMD CHANGE DUPLEX

機 能 通信方式（全二重/半二重方式）を切り換えます.

書 式 CMD CHANGE DUPLEX 〈ポート番号〉〔,〈通信方式〉〕

文 例 CMD CHANGE DUPLEX 1,1

解 説 通常，BASICは全二重通信方式でRS-232Cポートを制御します.

ところが，PC-TL101オートホン，PC-9863モデムボードあるいはDATAX ITM1200に内蔵されているモデムは1200ボーの転送速度においては半二重方式による通信しか行えません.

したがって，これらのモデムを介して1200ボーの通信を行う場合，コミュニケーションポートをオープンする前にシステムの通信方式を半二重方式に切り換えてやらなければなりません.

1200ボーでターミナルモードを使用する場合も同様です.

CMD CHANGE DUPLEXコマンドはシステムの通信方式を変更します.

〈ポート番号〉 1：RS-232C標準ポート（第1回線）に対する通信方式を変更します.
　　　　　　 2：RS-232C拡張ポート（第2回線）に対する通信方式を変更します.
　　　　　　 3：RS-232C拡張ポート（第3回線）に対する通信方式を変更します.

〈通信方式〉 0または省略：全二重方式
　　　　　　 1　　　　　：半二重方式

【注意】(1) CMD CHANGE DUPLEXコマンドはコミュニケーションファイルのオープンあるいはTERMコマンドの実行の前に実行しなければなりません.

一度，このコマンドで通信方式を切り換えますと，再度この命令で通信方式を切り換えるまで，指定された通信方式が有効となります.

(2) PC-TL102，PC-9865あるいはITM1212を使用する場合，この命令を実行しないでください.

これらの機器においては，全二重方式による1200ボーの転送がおこなわれます.

サンプルプログラム

```
100 CMD LINE OPEN "COM1:"
110 INPUT "TEL.NO";TN$
120 INPUT "FILE NAME ";FN$
130 OPEN FN$ FOR INPUT AS #1
140 DEF SEG=&HA000
150 BPS=PEEK(&H3FE6) AND &HF
160 IF BPS=3 THEN DPM=0 ELSE DPM=1
170 CMD DIAL TN$,1
180 CMD CHANGE DUPLEX 1,DPM
190 OPEN "COM1:N81N" AS #2
200 WHILE NOT EOF(1)
210  LINE INPUT #1,A$
220  PRINT #2,LEN(A$)+2
230  PRINT #2,A$
240 WEND
250 PRINT #2,0
260 CLOSE
270 CMD CHANGE DUPLEX 1,0
280 CMD LINE CLOSE
290 END
```

CMD DIAL

機　能　電話をかけます.

書　式　1）CMD DIAL [[#]〈電話機番号〉,]〈電話番号〉[,〈機能〉]

　　　　2）CMD DIAL [[#]〈電話機番号〉,]〈短縮番号〉

文　例　CMD DIAL　#1,TN$,TNC

　　　　CMD DIAL　#1,21

解　説

1）　指定された電話番号の自動発信を行い，指定されたモードの予約を行います.

〈電話番号〉は20桁以内の文字列で以下の値を指定できます.

0～9，＊，#…ダイアルコード

：…ポーズ

電話機は“：”の位置までダイアルした後，約2秒程度発信を中断し，その後残りの番号をダイアルします.

これは，構内交換機（PBX）から外線にダイアルする時，外線への接続要求を行う必要がある場合等に使用します.

(,)，－…セパレータ

番号を見やすくするもので特に意味はありません.

以下の電話番号はいずれも同じ動作をします.

034528000

03(452)8000

03-452-8000

(03)(452)(8000)

@…再ダイアル

直前にダイアルした電話番号に再度電話をかけます.

ただし@と他の文字を同時に指定することはできません.

また，本体の電源をOFFにしたりリセットスイッチを押したりした場合にはその直後に“@”を使用することはできません.

〈機能〉は自動発信後の動作を指定する数値であり以下のいずれかを指定します.

0または省略…ダイアル後通話モード

ダイアル後はただちに次の命令を実行しますので，以降は電話機を手動で操作してください.

・話し中の場合や相手が出ない場合には電話を切ります.

・相手が出た場合には通話を行い，その後電話を切ります.

1…アンサトーン検出後データ通信モード

呼び出し相手からアンサトーンが返されるまで待っています．一定時間待ってもアンサトーンが検出されない場合にはエラーとなります.

呼び出し相手からアンサトーンが返された場合にはデータ通信モードとなり，この命令の実行を終了します.

2…接続後データ通信モード

呼び出し相手が出るまで待っています．一定時間待っても相手が出ない場合にはエラーとなります．

呼び出し相手が出た場合にはデータ通信モードとなりこの命令の実行を終了します．

3…ダイアル後データ通信モード

ダイアル後はすぐデータ通信モードとなりこの命令の実行を終了します．

機能に1，2または3を使用する場合，相手はコンピュータ等の接続された自動着信可能な電話機が接続されていなければなりません．

もし手動着信された場合には正常に動作しない場合がありますので注意が必要です．

電話番号に再ダイアル“@”を指定し，機能を省略した場合には直前の機能が採用されます．機能を指定した場合には指定した機能が優先されます．

2）　〈短縮番号〉で指定された電話番号の自動発信を行います．

〈短縮番号〉は1～40までの数値でなければなりません．

短縮番号を指定した場合，その短縮番号に与えられている〈機能〉に関係なく，すぐ次の命令に制御が移行します．

この命令はオフラインコマンドモードでのみ使用可能です．

【注意】　オートダイアルの際，次のことに注意してください．

① 電話機の自動/手動スイッチを“自動”にしておきます（PC-9863モデムボードを使用する場合，この必要はありません）．

② 受話器は電話機に置いた状態にしておきます．

③ 電話機のモードスイッチがAUTOあるいはCALLにセッティングされていなければなりません．

④ 本体のボーレート指定と電話機の速度スイッチの値が一致していなければなりません．

⑤ PC-9863, PC-9865モデムボードでは短縮番号による自動発信はおこなえません．

サンプルプログラム

```
100 CMD LINE OPEN "COM1:" AS #1
110 INPUT "電話番号";TELNO$
120 IF TELNO$="" THEN 150
130 CMD DIAL #1,TELNO$,0
140 GOTO 110
150 CMD LINE CLOSE #1
160 END
```

CMD ERROR ON/OFF/STOP

機　能　通信エラーによる割り込みを許可/禁止/停止します．

書　式　1）CMD ERROR [[♯]〈電話機番号〉] ON

2）CMD ERROR [[♯]〈電話機番号〉] OFF

3）CMD ERROR [[♯]〈電話機番号〉] STOP

文　例　CMD ERROR　♯1 ON

解　説

1）　割り込み動作を許可します．

この命令実行後は，通信エラー割り込みが発生すると，CMD ON ERROR GOSUBによって定義された処理ルーチンに分岐します．

2）　割り込み動作を禁止します．

この命令実行後は，通信エラー割り込みが発生しても，処理ルーチンへの分岐は起こりません．

3）　割り込み動作を停止します．

この命令実行後は，通信エラー割り込みが発生しても割り込みが起こったことを覚えているだけで処理ルーチンへの分岐は起こりません．しかし，その後CMD ERROR ON命令によって割り込みが許可されると，その時点で処理ルーチンに分岐します．

ここで述べている通信エラーとはデータ通信モードのときモデムからデータを受信した際のエラーを意味しています．

サンプルプログラム

```
100 CMD LINE OPEN "COM1:" AS #1
110 CMD ON LINE #1 GOSUB *IL
120 CMD LINE #1 ON
130 GOTO 130
140 *IL
150 CMD RECEIVE #1,1
160 WHILE STATUS MODE(#1)<>2:WEND
170 OPEN "COM1:E71N" AS #1
180 CMD ON ERROR #1 GOSUB *IE
190 CMD ERROR #1 ON
200 FLN$="":A$=""
210 FOR I=1 TO 3000:NEXT
220 PRINT #1,CHR$(&H15);
230 IF LOC(1)=0 THEN 230
240 A$=INPUT$(1,1)
250 IF A$=CHR$(&HD) THEN 230
260 IF A$=CHR$(&HA) THEN 290
270 FLN$=FLN$+A$
280 GOTO 230
290 IF FLN$="" THEN 360
300 OPEN FLN$ FOR OUTPUT AS #2
310 IF LOC(1)=0 THEN 310
320 A$=INPUT$(1,1)
330 IF A$=CHR$(&H4) THEN 360
340 PRINT #2,A$;
350 GOTO 310
360 CLOSE
370 CMD MODE CUT #1
380 CMD RECEIVE #1,0
390 RETURN
400 *IE
410 PRINT "通信エラー発生":BEEP
420 DMY$=INPUT$(LOC(1),1)
430 RETURN 360
```

CMD LINE CLOSE

機　能　電話機を論理的に切り離します．

書　式　CMD LINE CLOSE [[#] 〈電話機番号〉]

文　例　CMD LINE CLOSE　#1

解　説　論理的に接続されているBASICとモデム内蔵電話機とを切り離し拡張電話制御命令の使用を終了します．

切り離し対象となるコミュニケーションポートをオープンしたファイルがある場合には，この命令の実行により自動的にそのファイルがクローズされます．

〈電話機番号〉はCMD LINE OPENで接続されている電話機番号を指定します．また〈電話機番号〉を省略すると現在接続されている電話機が全て切り離されます．

サンプルプログラム

```
100 CMD LINE OPEN "COM1:" AS #1
110 INPUT "電話番号";TELNO$
120 IF TELNO$="" THEN 150
130 CMD DIAL #1,TELNO$,0
140 GOTO 110
150 CMD LINE CLOSE #1
160 END
```

CMD LINE ON/OFF/STOP

機　能　着信による割り込みを許可/禁止/停止します．

書　式　1）CMD LINE [[#]〈電話機番号〉] ON

2）CMD LINE [[#]〈電話機番号〉] OFF

3）CMD LINE [[#]〈電話機番号〉] STOP

文　例　CMD LINE　#1 ON

解　説

1）　割り込み動作を許可します．

この命令実行後は，着信割り込みが発生すると，CMD ON LINE GOSUBによって定義された処理ルーチンに分岐します．

2）　割り込み動作を禁止します．

この命令実行後は，着信割り込みが発生しても，処理ルーチンへの分岐は起こりません．

3）　割り込み動作を停止します．

この命令実行後は，着信割り込みが発生しても割り込みが起こったことを覚えているだけで処理ルーチンへの分岐は起こりません．しかし，その後CMD LINE ON命令によって割り込みが許可されると，その時点で処理ルーチンに分岐します．

着信割り込みとは，相手から電話がかかってきたという事象を示しているものでありCMD RECEIVE命令（自動着信）の指定とは無関係に発生します．

【注意】　STATUS LINE関数を使用しているプログラムでは正常に着信割り込みが発生しない場合がありますのでCMD LINE ON/OFF/STOPと混在して使用しないでください．

サンプルプログラム

```
100 CMD LINE OPEN "COM1:" AS #1
110 CMD ON LINE #1 GOSUB *IL
120 CMD LINE #1 ON
130 GOTO 130
140 *IL
150 CMD RECEIVE #1,1
160 WHILE STATUS MODE(#1)<>2:WEND
170 OPEN "COM1:E71N" AS #1
180 CMD ON ERROR #1 GOSUB *IE
190 CMD ERROR #1 ON
200 FLN$="":A$=""
210 FOR I=1 TO 3000:NEXT
220 PRINT #1,CHR$(&H15);
230 IF LOC(1)=0 THEN 230
240 A$=INPUT$(1,1)
250 IF A$=CHR$(&HD) THEN 230
260 IF A$=CHR$(&HA) THEN 290
270 FLN$=FLN$+A$
280 GOTO 230
290 IF FLN$="" THEN 360
300 OPEN FLN$ FOR OUTPUT AS #2
310 IF LOC(1)=0 THEN 310
320 A$=INPUT$(1,1)
330 IF A$=CHR$(&H4) THEN 360
340 PRINT #2,A$;
350 GOTO 310
360 CLOSE
370 CMD MODE CUT #1
380 CMD RECEIVE #1,0
390 RETURN
400 *IE
410 PRINT "通信エラー発生":BEEP
420 DMY$=INPUT$(LOC(1),1)
430 RETURN 360
```

CMD LINE OPEN

機 能　電話機を論理的に接続します．

書 式　CMD LINE OPEN 〈ファイルディスクリプタ〉［AS［＃］〈電話機番号〉］

文 例　CMD LINE OPEN "COM1 :" AS ＃1

解 説　電話機と接続するコミュニケーションポートを宣言します．

この命令は電話機とBASICを論理的に接続する命令であり，全ての拡張電話制御命令の実行に先立って実行しなければなりません．

この命令を実行することにより初期モードからオフラインコマンドモードに移行します．

〈ファイルディスクプリタ〉は"COM :", "COM1 :", "COM2 :", "COM3 :"のいずれかを指定します．

ただし"COM2 :", "COM3 :"はRS-232C拡張インタフェースボードを実装している場合にのみ指定可能です．

〈電話機番号〉は1～3でなければなりません．省略すると1となります．

この命令は初期モードでのみ使用可能です．

サンプルプログラム

```
100 CMD LINE OPEN "COM1:" AS #1
110 INPUT "電話番号";TELNO$
120 IF TELNO$="" THEN 150
130 CMD DIAL #1,TELNO$,0
140 GOTO 110
150 CMD LINE CLOSE #1
160 END
```

CMD MODE CUT

機　能　電話を切ります.

書　式　CMD MODE CUT [[♯] 〈電話機番号〉]

文　例　CMD MODE CUT　♯1

解　説　電話を切ってオフラインコマンドモードに戻ります.
この命令はデータ通信モードで有効です.

サンプルプログラム

```
100 CMD LINE OPEN "COM1:" AS #1
110 INPUT "電話番号";TELNO$
120 INPUT "ファイル名";FLN$
130 OPEN FLN$ FOR INPUT AS #1
140 CMD DIAL #1,TELNO$,1
150 OPEN "COM1:E71N" AS #2
160 IF LOC(2)=0 THEN 160
170 IF INPUT$(1,2)<>CHR$(&H15) THEN 160
180 PRINT #2,FLN$
190 WHILE NOT EOF(1)
200 LINE INPUT #1,A$
210 PRINT #2,A$
220 WEND
230 PRINT #2,CHR$(&H4);
240 CLOSE
250 CMD MODE CUT #1
260 CMD LINE CLOSE #1
270 END
```

CMD ON ERROR GOSUB

機　能　通信エラーによる割り込みルーチンの開始行を定義します．

書　式　CMD ON ERROR [[＃]〈電話機番号〉] GOSUB〈行番号〉

文　例　CMD ON ERROR　＃1 GOSUB　＊TEL. REC

解　説　通信エラー割り込みが発生したときに分岐する処理ルーチンの開始行を定義します．〈行番号〉は割り込みが発生したときに分岐させる処理ルーチンの開始行を示します．処理ルーチンからの復帰は一般にサブルーチンの場合と同じでRETURN命令によって行います．単にRETURNとしたときには中断したところから再開し，後ろに行番号を指定したときにはその行から実行を再開します．

ここで述べている通信エラーとはデータ通信モードのときモデムからデータを受信した際のエラーを意味しています．

通信エラーの情報はコミュニケーションバッファ内に記憶されていますので割り込み処理ルーチン内でINPUT$関数等を使用してバッファ内のデータを読み取らないとエラー情報はリセットされません．

サンプルプログラム

```
100 CMD LINE OPEN "COM1:" AS #1
110 CMD ON LINE #1 GOSUB *IL
120 CMD LINE #1 ON
130 GOTO 130
140 *IL
150 CMD RECEIVE #1,1
160 WHILE STATUS MODE(#1)<>2:WEND
170 OPEN "COM1:E71N" AS #1
180 CMD ON ERROR #1 GOSUB *IE
190 CMD ERROR #1 ON
200 FLN$="":A$=""
210 FOR I=1 TO 3000:NEXT
220 PRINT #1,CHR$(&H15);
230 IF LOC(1)=0 THEN 230
240 A$=INPUT$(1,1)
250 IF A$=CHR$(&HD) THEN 230
260 IF A$=CHR$(&HA) THEN 290
270 FLN$=FLN$+A$
280 GOTO 230
290 IF FLN$="" THEN 360
300 OPEN FLN$ FOR OUTPUT AS #2
310 IF LOC(1)=0 THEN 310
320 A$=INPUT$(1,1)
330 IF A$=CHR$(&H4) THEN 360
340 PRINT #2,A$;
350 GOTO 310
360 CLOSE
370 CMD MODE CUT #1
380 CMD RECEIVE #1,0
390 RETURN
400 *IE
410 PRINT "通信エラー発生":BEEP
420 DMY$=INPUT$(LOC(1),1)
430 RETURN 360
```

CMD ON LINE GOSUB

機　能　着信による割り込みルーチンの開始行を定義します．

書　式　CMD ON LINE [[♯]〈電話機番号〉] GOSUB〈行番号〉

文　例　CMD ON LINE　♯1 GOSUB　＊TEL. REC

解　説　着信割り込みが発生したときに分岐する処理ルーチンの開始行を定義します．〈行番号〉は割り込みが発生したときに分岐させる処理ルーチンの開始行を示します．処理ルーチンからの復帰は一般にサブルーチンの場合と同じでRETURN命令によって行います．単にRETURNとしたときには中断したところから再開し，後ろに行番号を指定したときにはその行から実行を再開します．

着信割り込みとは，相手から電話がかかってきたという事象を示しているものでありCMD RECEIVE命令（自動着信）の指定とは無関係に発生します．

【注意】　STATUS LINE関数を使用しているプログラムでは正常に着信割り込みが発生しない場合がありますのでCMD ON LINE GOSUBと混在して使用しないで下さい．

サンプルプログラム

```
100 CMD LINE OPEN "COM1:" AS #1
110 CMD ON LINE #1 GOSUB *IL
120 CMD LINE #1 ON
130 GOTO 130
140 *IL
150 CMD RECEIVE #1,1
160 WHILE STATUS MODE(#1)<>2:WEND
170 OPEN "COM1:E71N" AS #1
180 CMD ON ERROR #1 GOSUB *IE
190 CMD ERROR #1 ON
200 FLN$="":A$=""
210 FOR I=1 TO 3000:NEXT
220 PRINT #1,CHR$(&H15);
230 IF LOC(1)=0 THEN 230
240 A$=INPUT$(1,1)
250 IF A$=CHR$(&HD) THEN 230
260 IF A$=CHR$(&HA) THEN 290
270 FLN$=FLN$+A$
280 GOTO 230
290 IF FLN$="" THEN 360
300 OPEN FLN$ FOR OUTPUT AS #2
310 IF LOC(1)=0 THEN 310
320 A$=INPUT$(1,1)
330 IF A$=CHR$(&H4) THEN 360
340 PRINT #2,A$;
350 GOTO 310
360 CLOSE
370 CMD MODE CUT #1
380 CMD RECEIVE #1,0
390 RETURN
400 *IE
410 PRINT "通信エラー発生":BEEP
420 DMY$=INPUT$(LOC(1),1)
430 RETURN 360
```

CMD RECEIVE

機　能　自動着信を許可/禁止します.

書　式　CMD RECEIVE [#〈電話機番号〉,] [〈機能〉]

文　例　CMD RECEIVE　#1

CMD RECEIVE 1

解　説　〈機能〉は自動着信後の動作モードを指定する数値であり以下のいずれかを指定します.

0または省略……自動着信を行わない

電話がかかってきた場合，電話機のベルが鳴り受話器を取るのを待っています.

1………着信後データ通信モードに切り換える.

電話がかかってきた場合，アンサートーンを返した後にデータ通信モードとなります.

着信割り込み処理ルーチン内でこの命令を実行するとただちにアンサートーンを返しデータ通信モードとなります.

【注意】　着信を検出してからデータ通信モードに移行するまでには時間がかかります. STATUS MODE関数を使用しデータ通信モードに移行したことを確認した後，データ通信を行ってください.

この命令はオフラインコマンドモードでのみ有効です.

サンプルプログラム

```
100 CMD LINE OPEN "COM1:" AS #1
110 CMD ON LINE #1 GOSUB *IL
120 CMD LINE #1 ON
130 GOTO 130
140 *IL
150 CMD RECEIVE #1,1
160 WHILE STATUS MODE(#1)<>2:WEND
170 OPEN "COM1:E71N" AS #1
180 CMD ON ERROR #1 GOSUB *IE
190 CMD ERROR #1 ON
200 FLN$="":A$=""
210 FOR I=1 TO 3000:NEXT
220 PRINT #1,CHR$(&H15);
230 IF LOC(1)=0 THEN 230
240 A$=INPUT$(1,1)
250 IF A$=CHR$(&HD) THEN 230
260 IF A$=CHR$(&HA) THEN 290
270 FLN$=FLN$+A$
280 GOTO 230
290 IF FLN$="" THEN 360
300 OPEN FLN$ FOR OUTPUT AS #2
310 IF LOC(1)=0 THEN 310
320 A$=INPUT$(1,1)
330 IF A$=CHR$(&H4) THEN 360
340 PRINT #2,A$;
350 GOTO 310
360 CLOSE
370 CMD MODE CUT #1
380 CMD RECEIVE #1,0
390 RETURN
400 *IE
410 PRINT "通信エラー発生":BEEP
420 DMY$=INPUT$(LOC(1),1)
430 RETURN 360
```

CMD STORE DIAL

機　能　短縮ダイアルを電話機に記憶させます．

書　式　CMD STORE DIAL [[#]〈電話機番号〉,]〈短縮番号〉
　　AS〈電話番号〉[,〈機能〉]

文　例　CMD STORE DIAL　#1, 21 AS "0423(64)1111"

解　説　電話番号を電話機に記憶させます．

〈短縮番号〉は21～40までの数値であり合計20個記憶できます．

〈電話番号〉は20桁以内の文字列であり記憶する番号を意味します．指定できる値は以下の通りです．

　0～9，*，#　……ダイアルコード
　：　　　　　……ポーズ
　(,)，－　　　……セパレータ

また，電話番号に空（NULL）ストリングを指定するとその短縮ダイアルに記憶されている内容がクリアされます．

〈機能〉は自動発信後の動作を指定する数値であり以下のいずれかを指定します．

　0または省略……ダイアル後通話モードとなる
　1……アンサートーン検出後データ通信モードとなる
　2……接続後データ通信モードとなる
　3……ダイアル後データ通信モードとなる

上記〈電話番号〉および〈機能〉の詳細はCMD DIALを参照してください．

この命令はオフラインコマンドモードでのみ使用可能です．

【注意】　PC-9863, PC-9865モデムボードでは，この命令を使用することはできません．

サンプルプログラム

```
100 CMD LINE OPEN "COM1:" AS #1
110 FOR I=21 TO 32
120 READ TELNO$
130 CMD STORE DIAL #1,I AS TELNO$,0
140 NEXT
150 CMD LINE CLOSE #1
160 END
170 DATA 03(454)1111,03(456)3111,011(231)0161,0222(61)5511
180 DATA 045(662)1621,0425(26)0911,011(231)0161,0762(23)1621
190 DATA 06(231)3111,082(247)4111,0878(72)4141,092(713)5151
```

STATUS DIAL

機　能	電話機に記憶されている電話番号の機能情報を調べます．
書　式	STATUS DIAL（[[#]〈電話機番号〉,]〈短縮番号〉）
文　例	STATUS DIAL（#1, 21）

解　説　電話機に記憶されている〈短縮番号〉に対応する自動発信後のモードを返します．

〈短縮番号〉は1～40までの数値でなければなりません．

自動発信後のモードは次のとおりです．

0：ダイアル後通話モードとなる

1：アンサートーン検出後データ通信モードとなる

2：接続後データ通信モードとなる

3：ダイアル後データ通信モードとなる

各値の意味の詳細についてはCMD DIAL命令を参照してください．

〈短縮番号〉に対応する電話番号が記憶されていない場合には－1が返されます．

この関数はオフラインコマンドモードでのみ使用可能です．

【注意】　PC-9863, PC-9865モデムボードでは，この関数を使用することはできません．

サンプルプログラム

```
100 CMD LINE OPEN "COM:"
110 FOR I=21 TO 30
120 PRINT I;STATUS DIAL$(I);"Func.";STATUS DIAL(I)
130 NEXT
140 CMD LINE CLOSE
```

STATUS DIAL$

機　能　電話機に記憶されている電話番号を調べます.

書　式　STATUS DIAL$ ([[#]〈電話機番号〉,]〈短縮番号〉)

文　例　STATUS DIAL$ (#1, 21)

解　説　電話機に記憶されている〈短縮番号〉に対応する電話番号を返します.

〈短縮番号〉は1～40までの数値でなければなりません.

〈短縮番号〉に対応する電話番号が記憶されていない場合には空(NULL)ストリングが返されます.

この関数はオフラインコマンドモードでのみ使用可能です.

【注意】　PC-9863, PC-9865モデムボードでは, この関数を使用することはできません.

サンプルプログラム

```
100 CMD LINE OPEN "COM:"
110 FOR I=21 TO 30
120 PRINT I;STATUS DIAL$(I)
130 NEXT
140 CMD LINE CLOSE
```

STATUS ERROR

機　能	通信エラーの有無を調べます.
書　式	STATUS ERROR [([#]〈電話機番号〉)]
文　例	STATUS ERROR (#1)

解　説　モデムからデータを受信したときエラーが発生したか否かを調べます.

エラーの状態は0～7の数字で返され，それぞれ次のエラーの状態を示します.

0：エラーなし

1：パリティー・エラー

2：オーバーラン・エラー

3：オーバーラン・エラー＋パリティ・エラー

4：フレーミング・エラー

5：フレーミング・エラー＋パリティ・エラー

6：フレーミング・エラー＋オーバーラン・エラー

7：フレーミング・エラー＋オーバーラン・エラー＋パリティ・エラー

この関数はデータ通信モードでのみ使用可能です.

サンプルプログラム

```
100 CMD LINE OPEN "COM1:" AS #1
110 IF STATUS LINE(#1) THEN GOSUB *IL
120 GOTO 110
130 *IL
140 CMD RECEIVE #1,1
150 WHILE STATUS MODE(#1)<>2:WEND
160 OPEN "COM1:E71N" AS #1
170 FLN$="":A$=""
180 FOR I=1 TO 3000:NEXT
190 PRINT #1,CHR$(&H15);
200 IF STATUS ERROR(#1)<>0 THEN *IE
210 IF LOC(1)=0 THEN 200
220 A$=INPUT$(1,1)
230 IF A$=CHR$(&HD) THEN 200
240 IF A$=CHR$(&HA) THEN 270
250 FLN$=FLN$+A$
260 GOTO 200
270 IF FLN$="" THEN 350
280 OPEN FLN$ FOR OUTPUT AS #2
290 IF STATUS ERROR(#1)<>0 THEN *IE
300 IF LOC(1)=0 THEN 290
310 A$=INPUT$(1,1)
320 IF A$=CHR$(&H4) THEN 350
330 PRINT #2,A$;
340 GOTO 290
350 CLOSE
360 CMD MODE CUT #1
370 CMD RECEIVE #1,0
380 RETURN
390 *IE
400 PRINT "通信エラー発生":BEEP
410 DMY$=INPUT$(LOC(1),1)
420 GOTO 350
```

STATUS LINE

機　能　着信があったかどうかを調べます．

書　式　STATUS LINE [([#]〈電話機番号〉)]

文　例　STATUS LINE (#1)

解　説　着信があったかどうかを調べる関数です．着信があった場合には真（－１），そうでない場合には偽（０）を返します．

この関数は現在の状態を調べるものではありませんので１回実行することにより着信があったという事象はクリアされます．

したがって着信割り込みを使用するプログラムでこの関数を使用すると正しく動作しないためCMD ON LINE GOSUB命令等と混在して使用しないでください．

サンプルプログラム

```
100 CMD LINE OPEN "COM1:" AS #1
110 IF STATUS LINE(#1) THEN GOSUB *IL
120 GOTO 110
130 *IL
140 CMD RECEIVE #1,1
150 WHILE STATUS MODE(#1)<>2:WEND
160 OPEN "COM1:E71N" AS #1
170 FLN$="":A$=""
180 FOR I=1 TO 3000:NEXT
190 PRINT #1,CHR$(&H15);
200 IF STATUS ERROR(#1)<>0 THEN *IE
210 IF LOC(1)=0 THEN 200
220 A$=INPUT$(1,1)
230 IF A$=CHR$(&HD) THEN 200
240 IF A$=CHR$(&HA) THEN 270
250 FLN$=FLN$+A$
260 GOTO 200
270 IF FLN$="" THEN 350
280 OPEN FLN$ FOR OUTPUT AS #2
290 IF STATUS ERROR(#1)<>0 THEN *IE
300 IF LOC(1)=0 THEN 290
310 A$=INPUT$(1,1)
320 IF A$=CHR$(&H4) THEN 350
330 PRINT #2,A$;
340 GOTO 290
350 CLOSE
360 CMD MODE CUT #1
370 CMD RECEIVE #1,0
380 RETURN
390 *IE
400 PRINT "通信エラー発生":BEEP
410 DMY$=INPUT$(LOC(1),1)
420 GOTO 350
```

STATUS MODE

機　能　現在のモードを調べます．

書　式　STATUS MODE [([#]〈電話機番号〉)]

文　例　STATUS MODE (#1)

解　説　現在の電話機のモードを返します．

返される値とモードとの対応は以下の通りです．

0：初期モード（電話機が接続されていない状態）

1：オフラインコマンドモード

2：データ通信モード

3：通話モード

各モードの詳細に関しては「19.1.3　4つのモード」を参照してください．

サンプルプログラム

```
100 CMD LINE OPEN "COM1:" AS #1
110 IF STATUS LINE(#1) THEN GOSUB *IL
120 GOTO 110
130 *IL
140 CMD RECEIVE #1,1
150 WHILE STATUS MODE(#1)<>2:WEND
160 OPEN "COM1:E71N" AS #1
170 FLN$="":A$=""
180 FOR I=1 TO 3000:NEXT
190 PRINT #1,CHR$(&H15);
200 IF STATUS ERROR(#1)<>0 THEN *IE
210 IF LOC(1)=0 THEN 200
220 A$=INPUT$(1,1)
230 IF A$=CHR$(&HD) THEN 200
240 IF A$=CHR$(&HA) THEN 270
250 FLN$=FLN$+A$
260 GOTO 200
270 IF FLN$="" THEN 350
280 OPEN FLN$ FOR OUTPUT AS #2
290 IF STATUS ERROR(#1)<>0 THEN *IE
300 IF LOC(1)=0 THEN 290
310 A$=INPUT$(1,1)
320 IF A$=CHR$(&H4) THEN 350
330 PRINT #2,A$;
340 GOTO 290
350 CLOSE
360 CMD MODE CUT #1
370 CMD RECEIVE #1,0
380 RETURN
390 *IE
400 PRINT "通信エラー発生":BEEP
410 DMY$=INPUT$(LOC(1),1)
420 GOTO 350
```

19.1.6 拡張電話制御命令使用時の注意事項

(1) 拡張電話制御命令を使用しているときにはコミュニケーションポートに対してOUT命令を実行しないでください．動作を保証できません．

(2) データ通信モードのときBASICのエラーが発生したりSTOPキーを押してプログラムの実行が中断した場合には自動的に電話が切れます．

(3) PC-TL101 ,PC-TL102オートホン，PC-9863, PC-9865モデムボードまたはDATAX ITM1200, ITM1212以外の電話機を接続した場合の動作は保証できません．

(4) データ通信中に何等かの事情により通信ができなくなった場合，永久に相手からのデータを待っている状態に陥る可能性があります．無人運転を行うシステムでは時間監視等を行うことによって電話を切ることができるようにプログラムしてください．

(5) CMD LINE OPENでコミュニケーションポートを接続した場合，そのポートに対応するコミュニケーションファイルは必ずデータ通信モード内でOPEN/CLOSEを実行して下さい．オープンしたままデータ通信モードを抜け出し，再度データ通信モードとなるとコミュニケーションファイルに対する入出力命令が誤動作する可能性があります．

(6) 異なった種類の電話機を接続してデータ通信をおこなう場合，正常動作しなければ受信側のモード設定スイッチはANSモードに設定してください．また，送信側はAUTOあるいはCALLにセッティングします．同一機種を接続する場合，両者ともAUTOモードでかまいません．

19.1.7 エラーメッセージ

拡張電話制御命令に関するエラーメッセージとして以下のものがあります．[ ] 内はそのエラー番号を示しています．

Telephone set not open [170]

意味 電話機が論理的に接続されていない．

原因 CMD LINE OPEN 命令を実行していない電話機に対して他の拡張電話制御命令を実行した．

Telephone set already open [171]

意味 電話機は既に接続されている．

原因 既にCMD LINE OPENが実行された電話機に対して再びCMD LINE OPEN命令を実行した．

Bad telephone set number [172]

意味 電話機番号が不正である．

原因 1，2，3以外の値を電話機番号として指定している．

Bad telephone number [173]

意味 電話番号が不正である．

原因 電話番号として指定できない文字が指定されている．

Communication I/O error [174]

意味 電話機とコマンドレベルの入出力ができない．

原因 電話機の故障，電話機の初期設定不良，接続不良等により電話機に対するコマンドレベルの入出力ができない．

Line busy [175]

意味 呼び出し相手が話し中である．

原因 電話の呼び出し相手が話し中である．

Can't execute across mode [176]

意味 現在のモードではこの命令は実行できない．

原因 現在のモードでは実行できない命令/関数を実行しようとした．

Request denied [177]

意味 電話機に対するコマンドが受け入れられなかった．

原因 電話機に対してコマンドを送出しても電話機からそのコマンドを否定された．

RS-232C (2nd/3rd) board not ready [82]

意味 RS-232C拡張ボードが接続されていない．

原因 RS-232C拡張ボードが実装されていないのに電話機と標準以外のコミュニケーションポートをCMD LINE OPEN命令で接続しようとした．

19.1.8 電話管理ユーティリティ（tele.n88）

電話管理ユーティリティの概要

電話管理ユーティリティは次の４つの機能を持ちます．

① 電話帳管理機能
② オートダイアル通話機能
③ ファイル転送機能
④ ファイル受信機能

(1) 電話帳管理機能

オートダイアルを行うためには通話先の電話番号や名前を登録しておくための電話帳が必要です．

電話帳には，氏名，電話番号をはじめ住所，所属等のコメント情報を登録しておきます．

また，通話先がBBS（掲示板サービス）や各種のデータベースサービスである場合，接続のための通信条件を設定することもできます．

電話帳管理機能には，電話帳の作成および電話帳の保守を行うための４つの機能が用意されています．

① 発信相手の新規登録
② 氏名，電話番号，属性等の変更
③ 発信相手の削除
④ 電話帳の表示

(2) オートダイアル通話機能

電話帳に登録している通話先を選ぶだけで，自動的に電話をかけることができます．この機能をオートダイアル機能と呼びます．

通話先がBBS（掲示板サービス）や各種データベースの場合には，オートダイアル後，あらかじめ設定されている通信条件にしたがってターミナルモードへ移行します．ターミナルモード移行後は，相手サービスとの規約にしたがって会話を行ってください．

(3) ファイル転送機能

指定されたファイルを通話先に転送します．

エラーチェック付きのファイル転送機能ですので，確実にデータを送ることができます．なお，ファイル受信側は本ユーティリティのファイル受信機能でなければなりません．

(4) ファイル受信機能

上記ファイル転送機能で送られてくるファイルデータを受信し，ファイルに格納します．

電話管理ユーティリティの使用に際して

(1) 電話機の準備

「19.1.2 拡張電話制御命令の使用に際して」に従って，電話機の準備を行ってください．

なお，電話管理ユーティリティは，RS-232C標準ポートを固定的に使用しますので，本体と電話機の接続はRS-232C標準ポートを使用してください．

(2) 拡張電話制御命令の使用宣言

電話管理ユーティリティは拡張電話制御命令を使用し，オートダイアルの制御を行います．このため，拡張電話制御命令の使用宣言が必要となります．

「19.1.2 (3) 拡張電話制御命令の使用宣言」に従い，準備してください．

注意 電話管理ユーティリティは日本語表示を行います．
このため，高解像度ディスプレイが必要となります．

電話管理ユーティリティの操作

(1) 電話管理ユーティリティの起動

電話管理ユーティリティはシステムディスク内に“tele.n88”という名前で登録されています．

“tele.n88”を直接RUNコマンドで起動することもできますが，通常は次のように“menu”プログラムを使用してください（N88-日本語BASIC(86) システムディスクがドライブ1にセットされているとします）．

RUN “menu” ⏎

次のメニュー画面が表示されますので，最後の“電話管理ユーティリティ（tele.n88)”にカーサを位置付けた後，⏎ キーを押します．

```
【 ユーティリティ 】

  フロッピィディスクの初期化              (format.nip)
  フロッピィディスクのコピー              (backup.n88)
  ファイルのコピー                        (xfiles.n88)
  メモリスイッチの設定                    (switch.n88)
  ハードディスクの初期化                  (format.hd )
  文節辞書ファイル・メンテナンス          (dicmen.n88)
  利用者定義文字メンテナンス              (mkfont.n88)
  プログラム・オートスタートのセット      (setinf.n88)
  ＤＩＳＫ ＣＯＤＥのコピー               (sysgen.nip)
  ハードディスクの障害復旧                (recov .hd )
  ハードディスクのディレクトリ表示        (dir   .hd )
  ハードディスクのファイル退避／復旧      (backup.hd )
  ５”１Ｄ／２Ｄコンバーター              (DDconv.n88)
  電話管理ユーティリティ                  (tele  .n88)
  日本語入力環境条件の設定                (setup .n88)

《 ↑キー、↓キーで選択してＲＥＴＵＲＮキーを押すと起動されます。 》
《 説明が必要な場合は、選択したのちＨＥＬＰキーを押してください。 》
```

電話管理ユーティリティが起動され次のメニュー画面が表示されますので，カーサを必要な機能に位置付け ⏎ キーを押し，機能を選択します．

```
電話管理ユーティリティ

1. 電話帳管理

2. オートダイアル通話

3. ファイル転送

4. ファイル受信

5. 終了

↑↓キーで移動
<┘ キーで機能選択
```

(2) 電話帳ファイルの指定

ファイル受信機能以外の機能を使用する場合，電話帳ファイルが必要となります．

電話帳管理機能を選択した場合，処理に入る前に次のようなメッセージが表示されます．

```
電話管理ユーティリティ

電話帳ファイル(TBOOK.SYS)が存在するドライブのドライブ番号を入力して下さい。
なお電話帳ファイルを新規に作成したい場合、HELPキーを押して下さい。電話帳ファイル
の新規作成が行えます。
?
```

初めて電話管理ユーティリティを使用する場合，HELP キーを押して新規に電話帳ファイルを作成する必要があります.
電話帳ファイルは“TBOOK. SYS”の名前で，指定されたドライブに作成されます.

2回目以降は，ドライブ番号のみ指定すれば，そのドライブに存在する電話帳ファイルが使用されます.

万一，何らかの理由で電話帳ファイルの内容が破壊されてしまった場合，初めて電話管理ユーティリティを使用する場合と同様に，電話帳ファイルの新規作成をおこなってください.

電話帳ファイル“TBOOK. SYS”は，52セクタ分の容量を持ち，104人分の情報を登録することができます.

(3) 各機能の操作方法

電話管理ユーティリティは電話帳管理機能，オートダイアル機能，ファイル転送機能あるいはファイル受信機能のいずれを使用する場合でも，画面に表示されるメニューあるいはガイドメッセージに従い操作が可能なように配慮されています.

従って，ここでは特に操作法の解説は行いません.

各機能使用時の留意点

(1) 日本語入力の方法

電話帳に通話先を登録する際あるいは電話帳の内容を変更する際等には，日本語入力を行う必要があります.

たとえば，通話先の氏名は日本語で入力します.

```
発信相手の新規登録

氏名を入力して下さい。氏名は日本語(12文字以内)で入力します。
  ?

R〔かな〕
```

このような場合，自動的に日本語入力モードになります．

日本語入力の方法はその時のシステムの日本語入力環境に依存します．すなわち，システムの日本語入力環境が文節変換入力の環境にある場合，文節変換による日本語入力を行います．

日本語入力の方法はシステムの起動時に決まります．

(2)　電話番号

電話帳に登録する電話番号は16桁以内の数字，コロンおよびハイフンで指定します．

コロンは次の意味を持ちます．

電話機は“：”の位置までダイアルした後，約2秒程度発信を中断し，その後残りの番号をダイアルします．

これは構内交換機（PBX）から外線にダイアルする時，外線への接続要求を行う必要がある場合等に使用します．ハイフンは形式的な区切り記号であり，あってもなくても良いものです．

〈指定例〉

(1)	03-452-8000	034528000	03-4528000
(2)	0：03-452-8000	0：034528000	0：03-4528000
(3)	8-647-533	8647533	8647-533

(3)　オートダイアル

オートダイアル時，特に次の点に御注意ください．

① 電話機の自動/手動スイッチを“自動”にしておきます．

② 受話器は電話機に置いた状態にしておきます．

③ 電話機のモードスイッチは，AUTOあるいはCALLにセッティングしておきます．

④ 本体のボーレート指定と電話機の速度スイッチの値を一致させます．

(4)　BBS/データベース等へのアクセス

BBS（掲示板サービス）やデータベースサービスへの接続を行う場合の手順は次の通りです．

① 電話機に通話先を登録する際，“発信相手はBBSですか（y/n）？”の問いに対して，“y”を応答します．

② BBSアクセスのための通信条件をあらかじめ登録しておきます．

通信条件の指定はメニューから選ぶだけの簡単な操作で行えます．

BBSアクセス時の通信条件は各サービスによって異なりますので，対象とするBBS通信条件を良く調べた上，指定してください．

```
                      発信相手の新規登録

PC－VAN              XX-XXX-XXXX      東京

ターミナルモードに入る際の通信条件を指定します。
                        ←→キーで移動 <┘ キーで機能選択

通信方式        : 全二重         半二重
パリティ        : 偶数パリティ   奇数パリティ  パリティなし
データビット長  : 8ビット        7ビット
ストップビット長: 1ビット        2ビット
XON/XOFF        : 有             無
SI/SO           : 無             有
DEL受信処理     : BS             NULL
送信時行末      : CR             CR+LF
CR受信処理      : CR+LF          CR
```

（ここまでは電話帳管理機能を使用します）

③ オートダイアル機能を使用してオートダイアルします．

BBS属性を持つ電話番号が通話先に選ばれると，まず，転送速度の問い合わせが行われます．

BBSにアクセスします．転送速度を指定して下さい．
1．300ボー　2．1200ボー

アクセスするBBSサービスの規約に従って転送速度を指定します（一般的に300ボーが使用されています）．

300ボーなら1を入力します．

注意 (1) ここで電話機の速度スイッチが③で指定した速度と同じ速度にセットされているか確認してください．

(2) 1200ボーを選択する場合，次のことに注意してください．1200ボー全二重方式による通信がおこなえる機器は次の2機種です．

① PC-TL102
② ITM1212
③ PC-9865モデムボード

これ以外の機器を使用する場合，半二重方式による通信となりますが，一般のBBSサービスは半二重方式の通信をサポートしておりませんので御注意ください．

④ オートダイアルします．呼び出し音に続き「ピー」という連続音が確認できたら，電話機をデータモードにして下さい．

というメッセージが表示され，オートダイアルが行われます．

オートダイアルが始まったら受話器を取り，呼び出し音を確認します．ビジーの場合は受話器を置き STOP キーを押してください．

呼び出し音に続く「ピー」という連続音が確認できたら，接続成功です．ここで電話機を通話モードからデータモードにします（通話/データスイッチを押してください）．

⑤ **ターミナルモードへ移行します．ターミナルモードへの移行後は，対象BBSとの規約に従って会話を行って下さい．**というメッセージが表示され，画面がクリアされて，ターミナルモードへ入ります．

注意 ターミナルモードでの操作については，「第13章　ターミナルモード」を参照してください．

⑥ この後，対象BBSとの規約に従って会話を開始します．

まず，サービス開始を要求するメッセージを送ります．

このメッセージはサービスによりまちまちですので，各サービスの規約に従って送信してください．

サービスが開始されると，通常，ユーザIDやパスワード等の問い合わせが行われますので，順次，応答してください．

⑦ 会話が終了したら SHIFT ＋ STOP キーを押します．

ターミナルモードから抜けオートダイアルの画面に戻ります．

(5) ファイル転送機能

プログラムファイル（アスキー形式およびバイナリ形式），機械語ファイル，データファイルのすべてを転送することができます．

ファイル転送の手順は次の通りです．

① 電話管理ユーティリティの初期メニュー画面において，“ファイル転送”を選択します．

② ガイドメッセージに従って，データの転送速度，通信方式および転送するファイルを指定します．

```
                    ファイル転送

転送速度を指定して下さい。
  1.300ﾎﾞｰ  2.1200ﾎﾞｰ
? 1
転送するファイルのドライブ番号とファイル名を指定して下さい。
  ドライブ番号：1
  ファイル名  ：TEST3
```

データの転送速度は300ボーと1200ボーのいずれかが指定できます．

300ボー：１秒間に約30バイトのデータを送ることができます．

1200ボー：１秒間に約120バイトのデータを送ることができます．

一般的にファイル転送処理の場合，1200ボーを指定します．

1200ボーを指定した場合，通信方式を問い合せてきます．

・PC-9865モデムボード，PC-TL102あるいはITM1212のいずれかを使用している場合，全二重方式を指定してください．

・PC-TL101, PC-9863あるいはITM1200のいずれかを使用している場合，半二重方式を指定します．

注意 電話機の速度スイッチが，ここで指定した速度と同じ速度にセットされているかどうか必ず確認しておいてください．

③ この後，オートダイアル通話と同様に，電話帳が表示されますので，ファイルを送る相手側の電話番号を選択します．

オートダイアルで相手を呼び出します．

④ 電話が通じたら，相手にファイルを受けるための準備をさせます．

PC－VAN	XX-XXX-XXXX	東京	BBS
日本電気株式会社	XX-XXX-XXXX	本社	
北海道支社	XX-XXX-XXXX	北海道ブロック	
東北支社	XX-XXX-XXXX	東北ブロック	
東京支社	XX-XXX-XXXX	首都圏ブロック	
中部支社	XX-XXX-XXXX	中部ブロック	
北陸支社	XX-XXX-XXXX	北陸ブロック	
関西支社	XX-XXX-XXXX	関西ブロック	

↑↓キーで通話先のエントリにカーサを位置付けて下さい。通話先にカーサを位置付けたら↵キーを押して下さい。↑↓↵以外のキーを押すと次の画面に進みます。

ファイルを受ける側は次のような準備をします．

(1) 電話を受け取ったらすぐに，電話管理ユーティリティを起動します．以降(6)までの間，電話を切らないでください．

(2) ファイル受信機能を選択します．

(3) 発信側のデータ転送速度および通信方式を確認し，同じ条件を指定します（この時電話機の速度スイッチも同じ速度にセッティングします）．

(4) 電話機のモードスイッチをAUTOにセッティングします．

注意 ただし，送信側と受信側の電話機が異なる場合，受信側はANSモードにセッティングしてください．

(5) ファイルデータを格納する際のファイル名を指定します（発信側が送ってくるファイル名と同じ名前で登録してよければ [↵] キーを応答します）.

```
                        ファイル受信

発信側の転送速度を確認し、それと同じ転送速度を選んで下さい。
 1.300ﾎﾞｰ  2.1200ﾎﾞｰ
? 1

受信ファイルを格納するドライブを指定して下さい。
? 1

受信ファイルをドライブに格納する際にファイル名を変更したい場合、そのファイル名を
指定して下さい。
発信側から送られてくるファイル名と同じ名前で格納してよければ、<┘キー のみ応答し
て下さい。
? █
```

(6) 発信側に準備ができた旨伝え，電話器を通話モードからデータモードにした後，受話器を置きます.

```
                        ファイル受信

発信側の転送速度を確認し、それと同じ転送速度を選んで下さい。
 1.300ﾎﾞｰ  2.1200ﾎﾞｰ
? 1

受信ファイルを格納するドライブを指定して下さい。
? 1

受信ファイルをドライブに格納する際にファイル名を変更したい場合、そのファイル名を
指定して下さい。
発信側から送られてくるファイル名と同じ名前で格納してよければ、<┘キー のみ応答し
て下さい。
? TEST
発信側に準備ができたことを告げ，電話機をデータモードにして下さい。
```

⑤ 送信側は，受信側のキャリア音（ピーという連続音）を確認した後電話機をデータモードにし，受話器を置きます．ファイルの転送が始まります．

注意 必ず受信側のキャリア音（ピーという連続音）を確認してから電話機をデータモードにしてください．

(6) 通話異常時の原因と処置

① オートダイアル

オートダイアル時，無応答になったり，通信異常エラーが発生したりした場合，STOP キーを押して，オートダイアル処理を中断した後，もう一度オートダイアルをやり直してください．それでもうまくいかない場合，次の点を確認してください．

- 「19.1.2 (1) 電話機のスイッチ設定」および「19.1.2 (2) 電話機と本体の接続」にしたがって電話機が本体に正しく接続されているかどうか確認し直してください．特に電話機のモードスイッチがAUTOあるいはCALLにセットされているかどうか確認してください．
- 拡張電話制御命令の使用宣言は行われているか（switch.n88ユーティリティ（Basic mode（電話制御命令の使用））で確認してください）．
- 電話機が自動モードになっているか（手動モードではオートダイアルできません）．

② オートダイアルできるが通話できない。

- 電話機が通話モードになっているか（データモードでは通話できません）．

③ オートダイアル後，ターミナルモードにはなるが，BBSとの会話がうまくいかない．

- 電話機がデータモードになっているか．
- 対象BBSのキャリア音（ピーという連続音）を確認した後，データモードにしたか．
- 電話機の速度スイッチが，BBSアクセス時に指定した転送速度と同じ速度にセッティングされているか．
- 通信方式（全二重/半二重），データビット長（8ビット/7ビット），パリティチェックの有無等の通信条件が，対象BBSの規約通りになっているか．

④ オートダイアル後，通話まではうまくいったがファイル転送がうまくいかない．

- 受信側の電話機は正しく本体に接続されているか．
- 送信側と受信側の転送速度および通信方式は一致しているか．
- 送信側/受信側とも，電話機の速度スイッチが指定した値と同じ速度にセッティングされているか．
- 受信側のキャリア音（ピーという連続音）を確認してから送信側の電話機をデータモードにしたか．
- 電話機のモードスイッチがAUTOになっているか（AUTOでうまくいかない場合，送信側をCALL，受信側をANSにしてみてください）．

19.2　サウンド制御機能

本機はFM音源によるサウンド発生機構を標準装備しています．

サウンド発生機構は次のような特徴を持っています．

(1)　サウンド発生部にFM（Frequency Moduration）方式音源のLSIを使用していますのでダイナミックでクリアな音の発生が可能です．

(2)　N_{88}-BASIC(86) 言語レベルでの音楽演奏が可能です．所定のメモリスイッチをONにしますと，N_{88}-BASIC(86) 本体と拡張サウンド制御命令実行部がリンクされ，PLAY文，VOICE文等の拡張サウンド制御命令が使用できるようになります．拡張サウンド制御命令実行部はサウンドインタフェースボード上のROMに格納されています．

(3)　6重和音（FM音源：3声，SSG音源：3声）による音楽演奏が可能です．

(4)　8オクターブに渡る音域の発生が可能です．

(5)　FM音源3声に対して多種多様な基本音色が用意されており，特に音色を作り出すまでもなく，音色番号を指定するだけで，リアルな音色による音楽演奏が可能です．もちろん，独自の音作りも可能です．また，種々の効果音も用意されています．

(6)　FM音源3声に対しては，それぞれ別の音色を割り当てることができます．たとえば，ストリング系音色（バイオリン，チェロ等）による三重奏，あるいはベースとスネアドラムでリズムを切り，ブラス系の音でメロディを奏でるというようなことがいとも簡単に行えます．

(7)　ビブラート，トレモロ効果等の特殊な効果音制御が可能です．

(8)　バックグラウンド演奏が可能です．たとえば，グラフィック画面に絵を描きながら音楽演奏を行うというような並列処理が可能です．

(9)　外部オーディオ機器用の出力端子が用意されています．外部オーディオ機器（ラジカセ，オーディオアンプ/スピーカ等）に接続することにより，ダイナミックなサウンドを楽しむことができます．外部オーディオ機器を接続しない場合，本体内のアンプ/スピーカが使用されます．

19.2.1　拡張サウンド制御命令の使用準備

拡張サウンド制御命令はメモリスイッチSW4・2^3ビットがONの状態で使用可能となります．

このスイッチは初期状態がONになっておりますので，特にスイッチをOFFにしない限り拡張サウンド制御命令の使用が可能です．

注　意　(1)　このスイッチがOFFの状態で，拡張サウンド制御命令を実行しますと，“Syntax error”のエラーが発生します．

(2)　サウンド機能が使用可能な状態でシステムを起動しますと，使用できない状態でシステムを起動した場合と比べ，プログラム領域が約2KB減少します．

(3)　サウンド機能を使用しますと，わずかながらインタプリタの処理速度が低下します．

19.2.2　拡張サウンド制御命令の概要

(1)　演奏のための初期状態の設定

拡張サウンド制御命令では，演奏のための情報の大部分をPLAY文で指定します．

N_{88}-BASIC(86) インタプリタはPLAY文を実行する毎に，これらの情報をサウンドバッファと呼ばれる領域に格納します．その後，この領域から情報を引出しながら演奏を行っていきます．

このサウンドバッファは，利用者が必要な大きさだけ確保する必要があります．サウンドバッファの確保はPLAY ALLOC文でおこないます（PLAY ALLOC文は音色，音量，テンポ，音の高さ等の初期化もおこないます）．

(2) 音楽データの表現

音楽演奏に必要な音程，長さ，テンポなどの情報を指示する命令としてPLAY文があります．PLAY文では，6チャンネルの同時発音を可能にしています．

また，PLAY文ではMML（ミュージックマクロランゲージ）という表現を用いて，先に述べた基本的な情報の他に特殊効果や音色の変化など，多くの情報を与えることができます．

(3) 音色の設定や変更

69種82個の音色データが用意されています．また，利用者が自分の好みに合わせて音色を変化させたり，全く最初から自分で音色を作り出したりすることができるように音色操作のための機能も用意されています．VOICE文，VOICE COPY文，VOICE LFO文などを利用して音色データの設定，読出し，一時変更などが可能です．

(4) 特殊効果の付加

特殊な演奏効果として，音量や音程にゆらぎを与える効果（LFO効果）を用意しています．この効果のためのデータは音色データの一部として設定可能です．また，VOICE LFO文で一時的に変更することも可能です．

(5) サウンドハードウエア（シンセサイザLSI）の直接制御

サウンドハードウエアに使用されているシンセサイザLSI（YM-2203）の内部レジスタを操作してより細かな音創りや演奏を可能にする命令としてVOICE REG文が用意されています．

(6) 命令との並行動作

拡張サウンド制御命令をN_{88}-BASIC(86) の他の命令と並行して実行させることができます．たとえば，音楽を演奏させながら絵を描くといったことが可能です．また，長い曲を複数のPLAY文に分けて演奏させる時など，音楽がとぎれない様になめらかに演奏させることができます．このような場合ON PLAY (c,n) GOSUB文，PLAY ON/OFF/STOP文などを使用します．

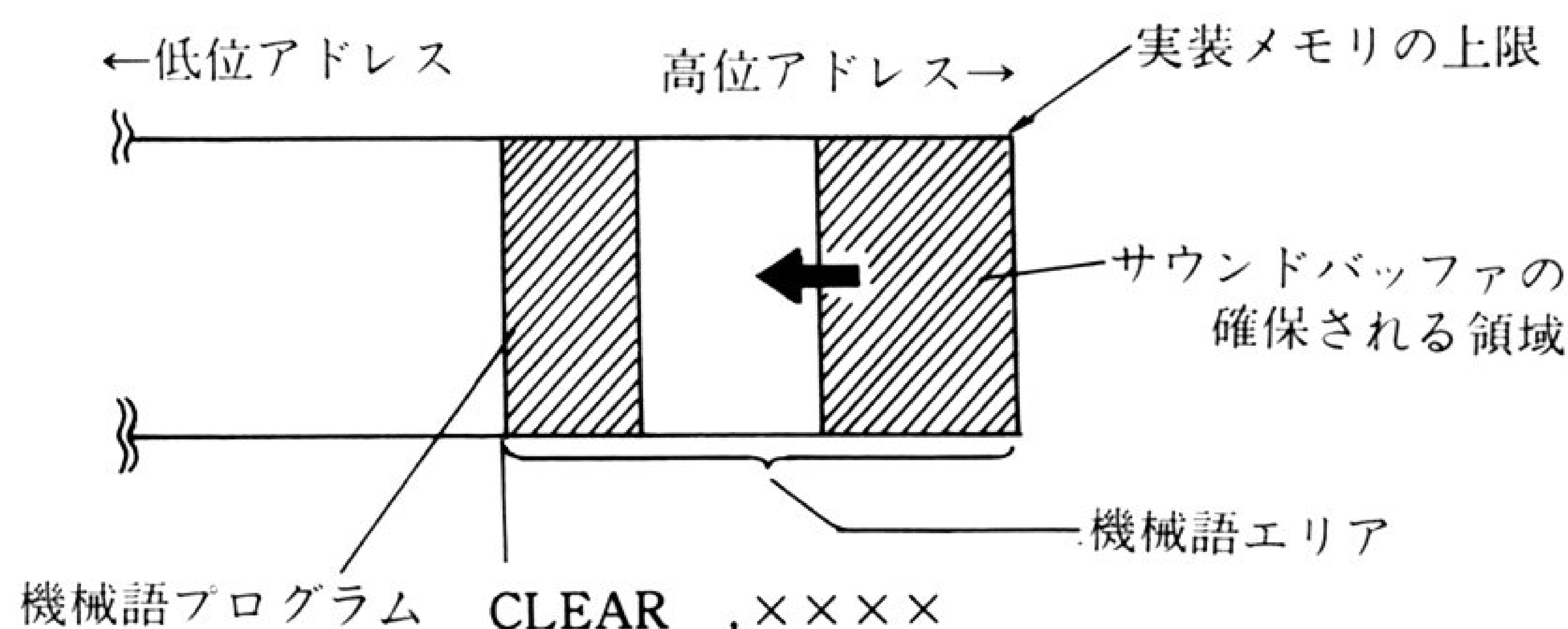
19.2.3 サウンドバッファについて

拡張サウンド制御命令は音楽情報を一時的に格納するための領域（サウンドバッファ）を必要とします．

このサウンドバッファはハードウエアに対する指示を一括してたくわえておく領域で，PLAY文で指示された音楽情報はこのバッファ領域に格納されます．

(1) サウンドバッファの確保される領域

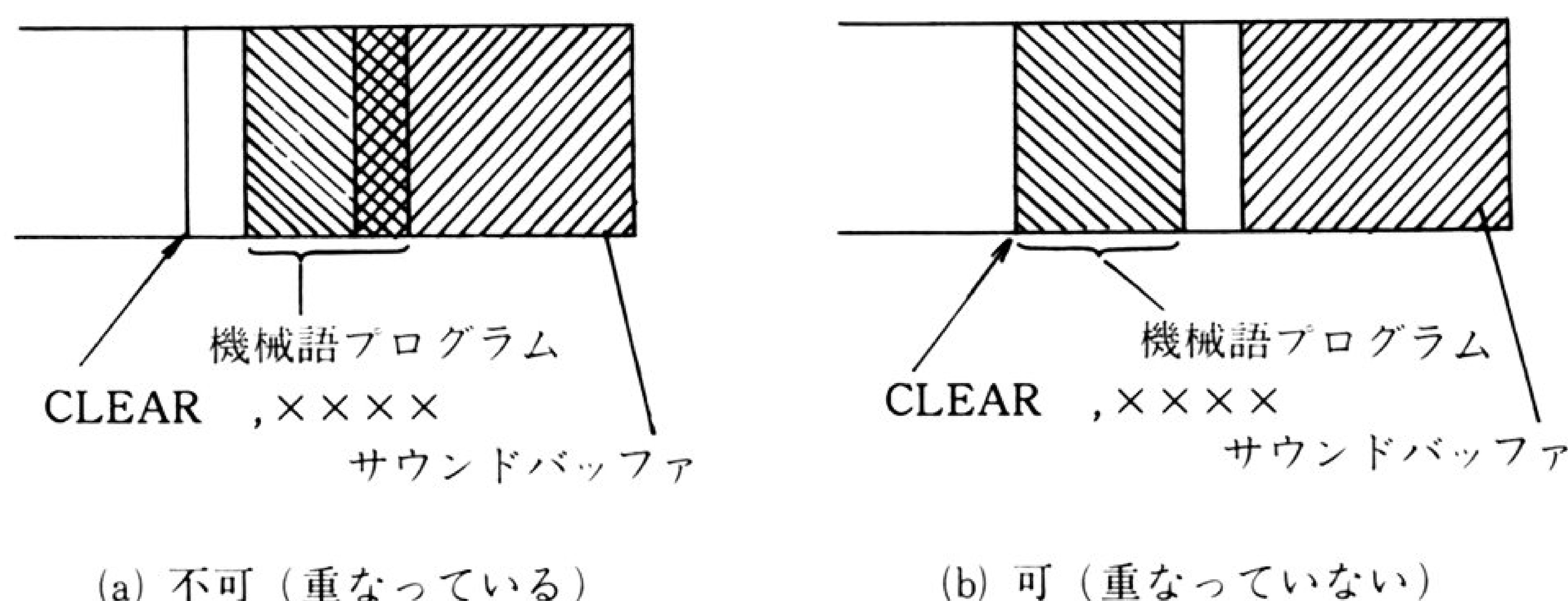
サウンドバッファはシステムが使用しない領域に確保されます．システムの使用するメモリの上限アドレスはCLEAR文の第2パラメータで宣言することができます（詳細はBASICリファレンスマニュアルを参照して下さい）．サウンドバッファは実装メモリの上限から低位アドレスに向かって確保されていきます．



(2)　サウンドバッファの確保

サウンドバッファの確保はPLAY ALLOC文でおこないます.

サウンドバッファはCLEAR文で指定したシステム空間の上限と実装メモリの上限の間に確保されます．サウンドバッファは実装メモリの上限から低位アドレスの方向に確保されますが，この領域はまた機械語領域としても使用されますので機械語プログラムを使用する場合は重ならないように注意して下さい．



(a) 不可（重なっている）　　(b) 可（重なっていない）

注　意　実装メモリの上限というのは，メモリスイッチSW 3で指定されたサイズです．したがって，メモリが640KB実装されていてもメモリスイッチSW 3で384KBしか指定されていなければ実装メモリサイズは384KBとなります．

(3)　サウンドバッファの変更

いったん確保したサウンドバッファは新たにPLAY ALLOC文が実行されるまで有効です．変更された後のサイズがCLEAR文により確保されている領域のサイズを越えるような場合は再びCLEAR文を実行し直す必要があります．

なお，PLAY ALLOC文でバッファのサイズを変更すると，演奏は中断され，初期状態に再設定されます．

(4)　サウンドバッファサイズの決定

確保するサウンドバッファのバッファサイズはメモリに余裕のある場合，プログラム中に存在するPLAY文のうち最も多い情報を必要とするPLAY文に合わせておくのがよいでしょう．

しかし，メモリに余裕がない場合には，一つの曲のデータをいくつかの部分に分割して表現する必要があります．バックグラウンドで（N_{88}-BASIC(86)のプログラムの実行と独立して演

奏する形態)，一つの曲を複数の部分に分割して，しかも，とぎれないように演奏するために種々の機能が用意されています．

ON PLAY (c,n) GOSUB, STATUS PLAYなどの命令は，サウンドバッファに残っている情報の量を知るための機能です．これらの命令により，とぎれずに次のPLAY文を実行させるタイミングをつかむことができます．

19.2.4 拡張サウンド制御命令

ここでは，拡張サウンド制御命令について詳しく解説していきます．

各命令の説明は次の構成で行われています．

命令の名前
- 機能
- 書式
- 文例
- 解説
- サンプルプログラム

機能 命令の機能を簡単に示します．

書式 命令の記述の仕方を示します．実際の入力時には次のような決まりに従ってください．

1．アルファベットの大文字で示された項目は，そのまま入力します．入力する場合は，小文字でも大文字でもかまいません．ただし，ダブルクォーテーション（ ” ）で囲まれた文字例（ファイル名以外）は，大文字と小文字を区別してください．

2．カギカッコ“<”，“>”で囲まれた項目は，ユーザーが指定します．

3．角カッコ“〔”,“〕”で囲まれた項目は，省略することができます．省略した場合，デフォルト値（BASICであらかじめ設定されている値）が適用されます．
以後のパラメータをすべて省略する場合は，コンマも含めて省略しますが，後に続くパラメータを指定する場合，それ以前のコンマはすべて指定する必要があります．
たとえば次のように指定します．

```
PLAY  ALLOC ,,,500
```

4．上記のカギカッコ，角カッコ以外の記号でカッコ（ ），コンマ，シャープ記号（#）などの記号は示された位置に正しく入力します．

文例 実際の入力の仕方の見本として簡単な例を示します．

解説 命令の使用方法や詳しい機能とそれに関して注意しなければならない点などを説明します．

サンプルプログラム いくつかの文を交えたプログラム例を示します．

PLAY ALLOC

機　能　**サウンドバッファの確保及び初期化を行います．**

書　式　PLAY ALLOC [<CH 1 バッファサイズ>] [, <CH 2 バッファサイズ>] [, <CH 3 バッファサイズ>] [, <CH 4 バッファサイズ>] [, <CH 5 バッファサイズ>] [, <CH 6 バッファサイズ>]

文　例　PLAY ALLOC 255, 255, 255

解　説　音楽演奏のためのサウンドバッファを各チャンネル毎に確保します．省略されたチャンネルのバッファサイズは 0 となります．

PLAY ALLOC 文は拡張サウンド制御命令を使用する前に必ず一度実行しなければなりません．

確保しようとするバッファサイズは必ずCLEAR文で確保した機械語エリアより小さくなければなりません．

PLAY ALLOC 文はサウンド機能を初期設定します．初期設定時の状態は次のとおりです．

SSGエンベロープ	：M255，S1（但し固定出力にされている）
SSG音量	：V7
音長	：L4（R4）
音の長さの割合	：Q7
オクターブ	：O4
テンポ	：T120
FMチャンネル音色	：0（デフォルト音色）
音色バンク	：立上時の状態に設定

サンプルプログラム

```
10 '***** PLAY SAMPLE PROGRAM *****
20 CLEAR ,&H9E00
30 PLAY ALLOC 255,255,255
40 PLAY "@25@V100L4CDEFGAB>C","@25@V100EFGAB>CDE","@25@V100GAB>CDEFG"
50 END
```

PLAY

機　能　**音楽演奏を行います.**

書　式　PLAY [#<モード番号>,] [<文字列1>] [, <文字列2>] [, <文字列3>] [, <文字列4>] [, <文字列5>] [, <文字列6>]

文　例　PLAY #0, "L4O5CDEFG", "L4O3GAB>CD"

解　説　文字列1～6によって与えられるMML（ミュージックマクロランゲージ）の指示に従って音楽を演奏します.

文字列1～3は，チャンネル1～3に対応するFM音源，文字列4～6は，チャンネル4～6に対応するSSG音源で，同時に6音までの同時演奏が可能です.

#<モード番号>は，0～2の値でチャンネル3のFM音源に対しどのようなモードで演奏するかを与えます.

#0：楽音モード　　（通常の演奏に適します）
#1：効果音モード　（効果音に適します）
#2：CSMモード　　（複合正弦波合成モード）

（各モードの詳細については，「19.2.6 より良い演奏のために」を参照して下さい）

<文字列n>はMMLです. MMLは次に示すようなコマンドを持っています.

(1) C,D,E,F,G,A,B　コマンド

音程をアルファベットで与えます（C：ド～B：シに対応します）. 音長の指定は"C4", "E16"などの様に音程のうしろに音長の数字で与えますが，省略可能です. 音長が省略された場合はLコマンド（後述）により指定された音長をとります.

(2) Mx　コマンド

SSG音源（チャンネル4～6）のエンベロープ周期を与えます. xの値は1～65535の範囲になくてはいけません. 初期値はM255です.

発生されるエンベロープ周期は，次の式により求められます.

$x = 667 \times T/256$　　T：エンベロープ周期（ms）
　　　　　　　　　　　　x：設定値

(3) Sx　コマンド

SSG音源（チャンネル4～6）のエンベロープ形状を指定します. xの値は1～15で，xの各々の値と対応するエンベロープ形状の対応は次図の通りです. 初期値はS1です.

X	エンベロープ形状	X	エンベロープ形状
0,1,2,3,9		11	
4,5,6,7,15		12	
8		13	
10		14	

SSG音源には固定音量出力モードとMxコマンドおよびSコマンドで指定される可変出力モードがありますが，各チャンネルを可変出力モードにするのはSxコマンドです．Mxコマンドは周期を設定するだけでモードは変えません．また，固定音量出力モードにするのは，後述するVコマンドです．

(4)　Vx　コマンド

各チャンネルに対し大まかに音量を設定します．

“V 0”が最小で“V15”が最大音量です．SSG音源に対する初期値は“V 7”です．FM音源については各音色ごとに最も適した音量が設定されるので初期値は存在しません．また，FM音源については音量の大小により音色も多少変化します．音色番号を切り換えると，@V-xコマンドおよびVxコマンドによる設定は無効になります．

(5)　Lx　コマンド

音長を省略した時の音の長さを設定します．

xは1～64の範囲で，1は全音符，8は8分音符といったように指定します．初期値は“L 4”です．

(6)　Qx　コマンド

音長により定められる長さ（ステップタイム）と，その間で実際に音を出している長さ（ゲートタイム）の比を設定します．xの値は1～8の範囲で，ゲートタイム/ステップタイムの比は1/8×xで求められます．初期値は“Q 7”です．

Qx	Q 1	Q 2	Q 3	Q 4	Q 5	Q 6	Q 7	Q 8
ゲートタイム／ステップタイム	1/8	2/8	3/8	4/8	5/8	6/8	7/8	8/8

(7) Ox　コマンド

オクターブ値を設定します．xの値は1～8で，初期値は“O 4”です．

(8) >，<　コマンド

現在のオクターブを1オクターブ上げ（>）たり下げ（<）たりします．

(9) Kx　コマンド

xで指定された高さの音を発生します．x（キー番号）と実際に発生される音程の対応は次の通りです．ただし，FM音源においては，“O 9 C”は“O 1 C”と同じになります．

オクターブ／音程	O 1	O 2	O 3	O 4	O 5	O 6	O 7	O 8	O 9
C	0	12	24	36	48	60	72	84	96
C♯	1	13	25	37	49	61	73	85	
D	2	14	26	38	50	62	74	86	
D♯	3	15	27	39	51	63	75	87	
E	4	16	28	40	52	64	76	88	
F	5	17	29	41	53	65	77	89	
F♯	6	18	30	42	54	66	78	90	
G	7	19	31	43	55	67	79	91	
G♯	8	20	32	44	56	68	80	92	
A	9	21	33	45	57	69	81	93	
A♯	10	22	34	46	58	70	82	94	
B	11	23	35	47	59	71	83	95	

⑽　Tx　コマンド

演奏するテンポを設定します．どのチャンネルでテンポを指定してもかまいません．ただし，全チャンネルを通してテンポはただひとつなので，最後に指定したチャンネルのテンポが有効となります．xの範囲は32～255までで，“T120”が初期値です．T120は“♩＝120”に相当します．

⑾　Rx,Px　コマンド

休符です．xの値は1～64で，音長を指定します．

⑿　＋（♯），－

いずれも音程（C,D,E,F,G,A,B）のあとにつけて半音階を表します．“＋”または“♯”で半音上げ，“－”で半音下げます．

⒀　．（ピリオド）

音長のあとにつけて，符点音符を表します．

⒁　＾

前後の音長をつないで一つにします．例えば，“C8＾8”のように表現される音は“C4”と等しくなります．

⒂　｛‥‥｝x

連符を表します．xで指定された音長を｛　｝の内の音符の個数で等分します．ただし，｛　｝の内で音長を指定した場合は，正しい連符となりません．また，連符の中に連符や間接指定を含める事もできません．

⒃　@xコマンド

xで示される音色番号の音に音色を切換えます．

xの範囲は0～81です．このコマンドはFM音源チャンネルについてのみ有効です．音色番号と音色の対応は初期状態で次のようになっています．

内蔵基本音色一覧表

音色番号	音色名	説明	備考
0	DEFAULT	(HARPSIC-11番と同じ)	
1	BRASS 2	低域の金管楽器	
2	STRING 2	高域の弦楽器	
3	EPIANO 3	丸みを持ったエレクトリックピアノ	
4	EBASS 1	シンセサイザベース	
5	EORGAN 1	堅めのエレクトリックオルガン	
6	PORGAN 1	低域のパイプオルガン	
7	FLUTE	フルート	O 5が最適です
8	OBOE	オーボエ	
9	CLARINET	クラリネット	
10	VIBRAPHN	ビブラフォン	
11	HARPSIC	ハープシコード	
12	BELL	ベル	
13	PIANO	アコースティックピアノ	
14	MUSHI	虫の鳴き声	
15	DESCENT	下降する様な効果音	
16	UFO	UFOが飛んでいる様な効果音	
17	GRANPRI	レーシングカーが走って行く様な効果音	
18	LASER 1	SF的な効果音　1	
19	LASER 2	SF的な効果音　2	
20	SINWAVE	正弦波	
21	BRASS 1	高域の金管楽器	
22	BRASS 2	低域の金管楽器	
23	TRUMPET	トランペット	O 5が最適です
24	STRING 1	低域の弦楽器	
25	STRING 2	高域の弦楽器	
26	EPIANO 1	エレクトリックピアノ	
27	EPIANO 2	堅めのエレクトリックピアノ	
28	EPIANO 3	丸みを持ったエレクトリックピアノ	
29	GUITAR	エレキギター	O 5が最適です
30	EBASS 1	シンセサイザベース	
31	EBASS 2	ウッドベースに近い音	O 3が最適です
32	EORGAN 1	堅めのエレクトリックオルガン	
33	EORGAN 2	やや丸いエレクトリックオルガン	
34	PORGAN 1	低域のパイプオルガン	

音色番号	音　色　名	説　　　　　　　明	備　　　考
35	PORGAN 2	高域のパイプオルガン	
36	FLUTE	フルート	O 5が最適です
37	PICCOLO	ピッコロ	
38	OBOE	オーボエ	
39	CLARINET	クラリネット	
40	GROCKEN	グロッケン	
41	VIBRAPHN	ビブラフォン	
42	XYLOPHN	シロフォン	
43	KOTO	琴	
44	ZITAR	チター	
45	CLAVINET	クラビネット	
46	HARPSIC	ハープシコード	
47	BELL	ベル	
48	HARP	ハープ	O 5が最適です
49	BELL/BRASS	ベルと金管楽器のユニゾン	
50	HARMONICA	ハーモニカ	
51	STEELDRUM	スチールドラム	
52	TIMPANI	ティンパニー	O 3が最適です
53	TRAIN	汽笛の音	
54	AMBULANCE	救急車の音	長めの音が適します
55	TWEET	小鳥の声	O 6，Q 4が最適です
56	RAIN DROP.	雨だれの音	長めの音が適します
57	HORN	ホルン	
58	SNARE DRUM	スネアドラム	
59	COW BELL	カウベル	
60	PERC 1	バネのはじけるような音	
61	PERC 2	水の入ったグラスをはじくような音	O 5が最適です
62	PERC 3	ウインドチャイムのような音	O 6が最適です
63	MUSIC BOX	オルゴールのような音	
64	CELLO	チェロ	
65	LOW BRASS	チューバのような低域の金管楽器	
66	WW ENSNBL	木管楽器のアンサンブル	
67	AC GUITAR	アコースティックギター	
68	FLUTE/HARP	フルートとハープのユニゾン	
69	FUNK PLUC	打弦楽器の音	
70	FUNK BASS	ファンキーなベースの音	
71	SYN LEAD	メロディーライン向のリードに適した音	

音色番号	音色名	説明	備考
72	METAL CLES	金属的な効果音 1	O 3が適します
73	STAIN	金属的な効果音 2	O 3が適します
74	CUBIN GLASS	氷がグラスの中を転がるような音	O 5が適します
75	HUMAN	人の声のような音	
76	WOOD BASS	ウッドベース	O 3が最適です
77	CHIMES	チャイム	O 5が最適です
78	SPACY	機械的な口笛の音	O 5が最適です
79	OBOE/B.CLAR	オーボエとバスクラリネットのユニゾン	
80	OLD STRING	古いレコードのバイオリンのような音	
81	STEEL'S CRY	金属的な泣き声のような音	

備考の欄に特に記入のないものは，どのような音域でも良いか，またはデフォルトが最も適していることを示します．

(17)　Yr,d　コマンド

シンセサイザLSIのレジスタに，直接，値を設定します．

rはレジスタ番号で0～178，dはレジスタ内容で0～255の範囲です．定義されていないレジスタに対するこのコマンドの動作の保証はありません．また，サウンド拡張命令がハードウエアの制御のために使用しているレジスタについて書込みを行った場合も，動作の保証はありませんので注意して下さい．

(18)　@Vx　コマンド

FM音源に対する音量を細かく設定します．

xは0～127の範囲で，0が最小音量，127で最大音量となります．

(19)　@Wx　コマンド

xで指定された長さだけ何もしないで待ちます．

@Wxコマンドは，Yr,dコマンドなどで音を発生させる場合に有効です（Rx,Pxコマンドでは，いったん発音を停止させてしまいますので音がとぎれてしまいます）．普通の音符（C,D,E,F,G,A,B等）のあとに@Wxが来た場合はRx,Pxコマンドと同等です．

(20)　__x（アンダーバー）

移調を指示します．xは音程で，A,B,C…のように指定します．xによって示される音程をCの音とみなし，以降のMMLを移調します．各チャンネルごとに独立して指定できます．ただしKxコマンドによる指定は移調しません．

移調は，次の規則により行われます．

○音程（A,B,C,D…）をキー番号（0～96）に直します．

○その値に，次表に示すようなベース値を加えて対応するキー番号の音を発生します．

指定調	G♭	G	A♭	A	B♭	B	C	C♯	D	D♯	E	F
ベース値	－6	－5	－4	－3	－2	－1	0	1	2	3	4	5

例えば，次のような指定があった場合

PLAY“__DDEFG”

実際に発生される音は

PLAY“EF♯GA”

と等しくなります．

(21)　！　コマンド

LFO効果（ビブラート，トレモロ）を加えます．音色番号指定直後や，VOICE LFO文実行直後はLFO効果が自動的に与えられていますが，＊コマンドでLFO効果を解除した場合，！コマンドでLFO効果を再設定することができます（ただし，音色データの値によっては，LFO効果がかからない場合があります）．

(22)　＊　コマンド

！コマンドの逆でLFO効果を切ります．

(23)　Zp,v　コマンド

指定チャンネルの音色パラメータの内容を設定します．

pはパラメータ番号で，0～49の範囲です．また，Vはパラメータの内容で0～16383までの値です．

pのパラメータ番号は，VOICE文やVOICE COPY文における音色用配列の添字に次のように対応します．

p＝（第1添字）＋（第2添字）×5

例えば，音色用配列（3，1）に対応するパラメータ番号は

3＋1×5＝8

となります．

Vの値については各音色パラメータの有効範囲外の値を設定しようとした場合，動作は保証しません．また，負の値は8ビットの符号付数値に直して与えて下さい．例えば，－5を与えたい時は，256をその値に加えて下さい．

256＋（－5）＝251

として与えます．

(24) MF,MB　コマンド

PLAY文の演奏形態を指定します．

“MF”（フォアグラウンド演奏指定）は1つのPLAY文による演奏が終了してから次の命令の実行へ移ります（初期状態は“MF”の状態になっています）．“MB”（バックグラウンド演奏指定）はPLAY文の音楽情報をサウンドバッファに格納してすぐに次の命令に移ります．サウンドバッファに格納された情報はその後順次ハードウエアにより引出されていきます．バックグラウンド演奏指定により，複数のPLAY文に分割された曲をなめらかにつなげていくことができます．“MF”，“MB”ともに各チャンネルごとに指定できますが，“MF”で指定されたチャンネルと“MB”で指定されたチャンネルが同時にあった場合，“MF”指定のチャンネルが終了するまで次の文へは移れません．

“MB”が指定された場合，次の条件により演奏は中断されます．

○STOPキー又はSTOP文の実行

○PLAY ALLOC文の実行

○エラーの発生による中断

(25) MMLの変数指定

MMLの一部を文字変数や数値変数で置き換えることも可能です．このことを間接指定と言います．

○数値の置き換え

“L＝A;”のように“＝”＋(変数名)＋“；”で表わします．ただし，変数名は配列の要素であったり型宣言文字（“！”や“％”等）がついていてはいけません．

○文字列の置き換え

“X＝A＄;”のように“X＝”＋(文字変数名)＋“；”で表わします．ただし，変数名は配列の要素であってはなりません．

注　意　(1)　エラー発生時には，音色は自動的に0番に設定されます．

(2)　PLAY文により演奏される音長は他の処理等の影響により若干長くなることがあります．

(3)　“MB”（バックグラウンド演奏指定）による演奏時，BASICプログラムの実行は若干遅くなります．

(4)　“MF”（フォアグラウンド演奏指定）による演奏時，連続してPLAY文を実行

させた場合，命令解析の時間のためにMMLの最後の音長が長めになったり，演奏が息つぎをしている様に聞こえる事があります．

(5)　音量が大きすぎると音がひずむ恐れがあります．また，2CH以上を使用し演奏をおこなう場合，1CHのみで演奏する場合に比べ音量が若干大きくなります．このような場合，@VxコマンドあるいはVxコマンドにより音量を調整してください．

サンプルプログラム

```
10 '***** PLAY SAMPLE PROGRAM *****
20 CLEAR ,&H9E00
30 PLAY ALLOC 255
40 FOR I=0 TO 81
50 PRINT "音色=";MID$(STR$(I),2),
60 PLAY "T80O4L8@=I;@V120C1" : FOR J=1 TO 1000 : NEXT
70 NEXT I
80 END
```

VOICE

機 能 **音色バンクを定義します**

書 式 VOICE＜音色番号＞，＜音色用整数型配列名＞

解 説 VOICE 5,A%

解 説 ＜音色番号＞で示される音色番号の音色パラメータを＜音色用整数型配列名＞で与えられる配列の音色パラメータにします．＜音色番号＞は1～81の範囲の値です．

（音色バンクについて）

拡張サウンド制御命令では81種類の音色を使用することができます．これらの音色は番号で管理されており，初期状態では内蔵音色のデータが各音色番号に割当られています．この音色番号と音色データのことをまとめて音色バンクと呼びます．音色バンクは0番のデフォルト音色以外は全てユーザが再定義して自分の好みの音色を割当てることができます．再定義された音色はPLAY ALLOC文やCLEAR文，VOICE INIT文が実行されるまで有効です．

配列により与えられる音色パラメータは次のように対応しています（詳細については「19.2.6 より良い演奏のために」を参照して下さい）．

配列名をVOICE%としますと，

VOICE%（0,0）：フィールドバック/アルゴリズム
VOICE%（0,1）：オペレータマスク
VOICE%（0,2）：LFO波形
VOICE%（0,3）：LFO SYNCディレイ
VOICE%（0,4）：LFO速さ
VOICE%（0,5）：LFOピッチ変調深さ（微調整）
VOICE%（0,6）：LFO振幅変調深さ（微調整）
VOICE%（0,7）：LFOピッチ変調深さ（粗調整）
VOICE%（0,8）：未使用
VOICE%（0,9）：未使用
VOICE%（1,x）～（4,x）：各オペレータに対するパラメータを与えます．
VOICE%（OP,0）：アタック係数
VOICE%（OP,1）：ディケイ係数
VOICE%（OP,2）：サスティン係数
VOICE%（OP,3）：リリース係数
VOICE%（OP,4）：サスティンレベル
VOICE%（OP,5）：出力レベル
VOICE%（OP,6）：キーボードレイトスケーリング深さ

VOICE%（OP,7）：マルチプル

VOICE%（OP,8）：デチューンレイト

VOICE%（OP,9）：LFO振幅変調深さ

注　意 (1) VOICE文，VOICE COPY文で使用する配列は4×9の大きさで，必ずOPTION BASE 0を宣言して使用して下さい．

(2) VOICE文で配列を使用した後はERASE文を実行しないで下さい．ERASEを行った後のサウンド拡張命令の動作は保証しません．ERASEを行うには，PLAY ALLOC文，VOICE INIT文などを実行し，配列をサウンド拡張命令から解放した後に行います．

VOICE COPY

機　能　音色バンクの音色パラメータをコピーします.

書　式　VOICE COPY<音色番号>, <音色用整数型配列>

文　例　VOICE COPY 4, PARA%

解　説　<音色番号>は0～81の範囲の値で, 指定された音色番号の音色パラメータを<音色用整数型配列>にコピーします. 配列の内容はVOICE文と同じ構成になっています.

サンプルプログラム

```
10 REM ********** VOICE sample program **********
20 CLEAR ,&H9D00
30 DIM A%(4,9)
40 PLAY ALLOC 255
50 PLAY "O4@3@V110CDEFGAB>C"
60 VOICE COPY 3,A%
70 VOICE 2,A%
80 PLAY "O4@2@V110CDEFGAB>C"
90 VOICE INIT
100 PLAY "O4@2@V110CDEFGAB>C"
110 END
```

VOICE LFO

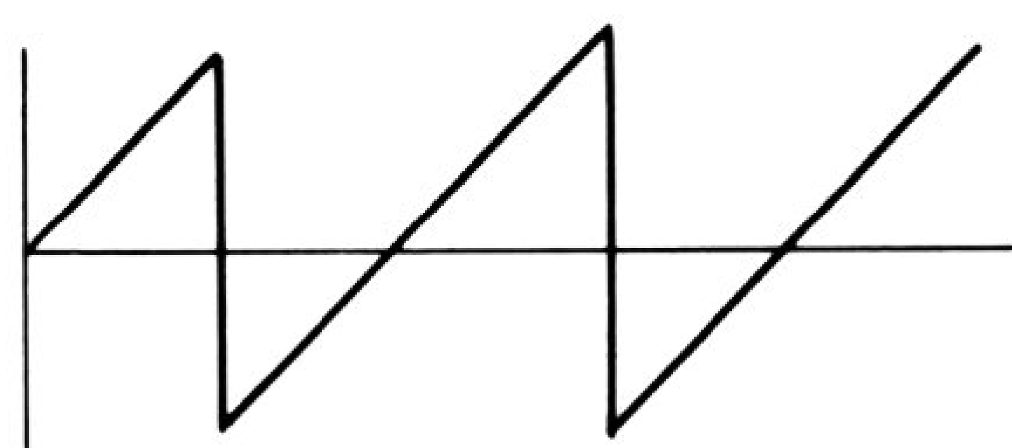
機　能　**各チャンネルの出力にLFO効果を与えます.**

書　式　VOICE LFO **＜チャンネル番号＞** [, **＜波形＞**] [, **＜SYNCディレイタイム＞**] [, **＜速さ＞**] [, **＜ピッチ変調深さ（微調整）＞**] [, **＜振幅変調深さ（微調整）＞**] [, **＜ピッチ変調深さ（粗調整）＞**]

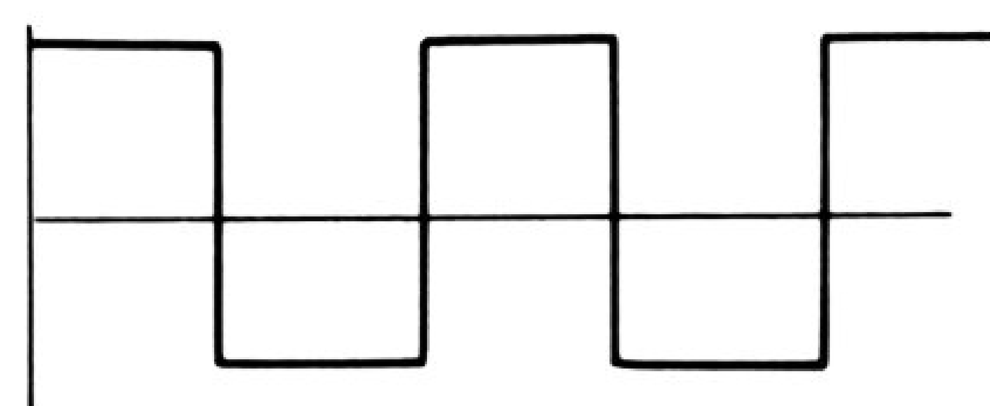
文　例　VOICE LFO 2, 3, 10, 1500, −15, 0, 7

解　説　＜チャンネル番号＞で指定されたチャンネルの音にビブラート，トレモロなどのLFO効果をつけます.

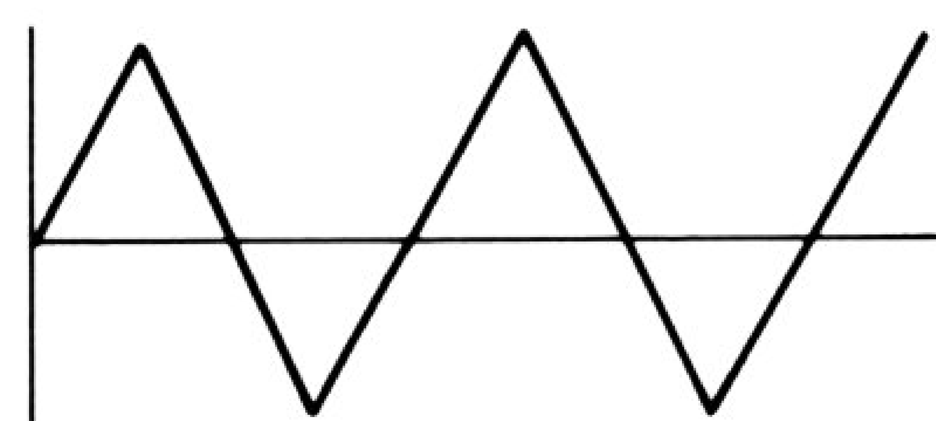
＜波形＞は0～5の範囲でLFOの変調波形を次の中から選びます.



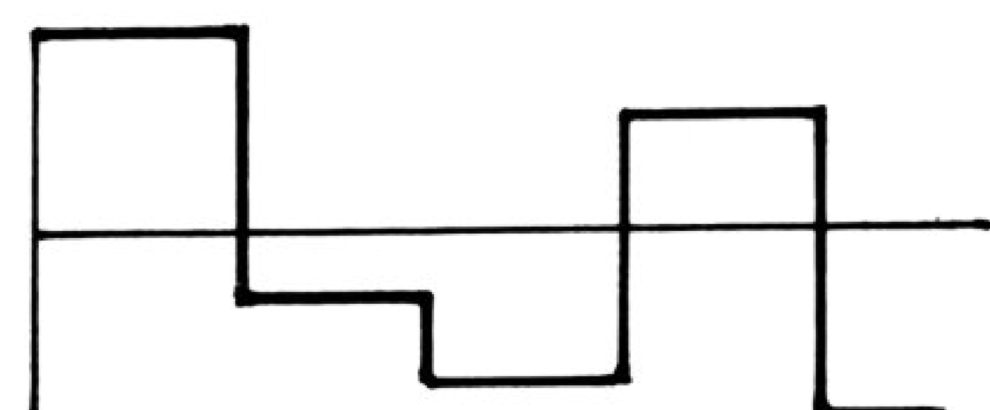
0：ノコギリ波



1：矩形波



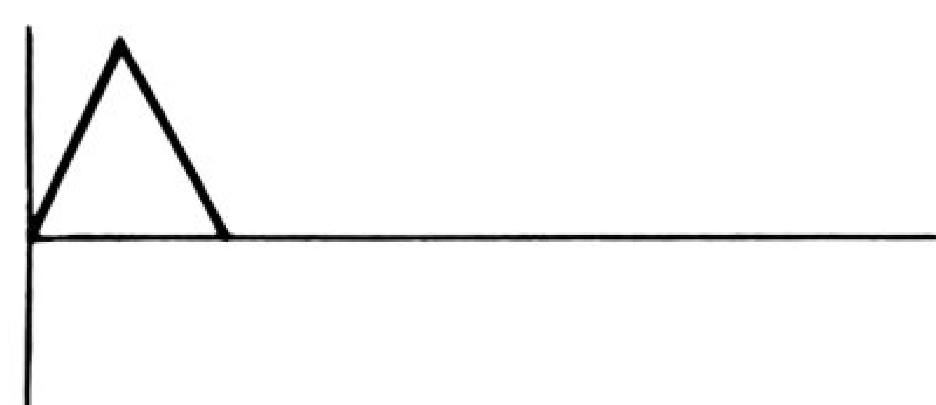
2：三角波



3：サンプル＆ホールド
（一定周期でランダムに変化）



4：ノコギリ波ワンショット

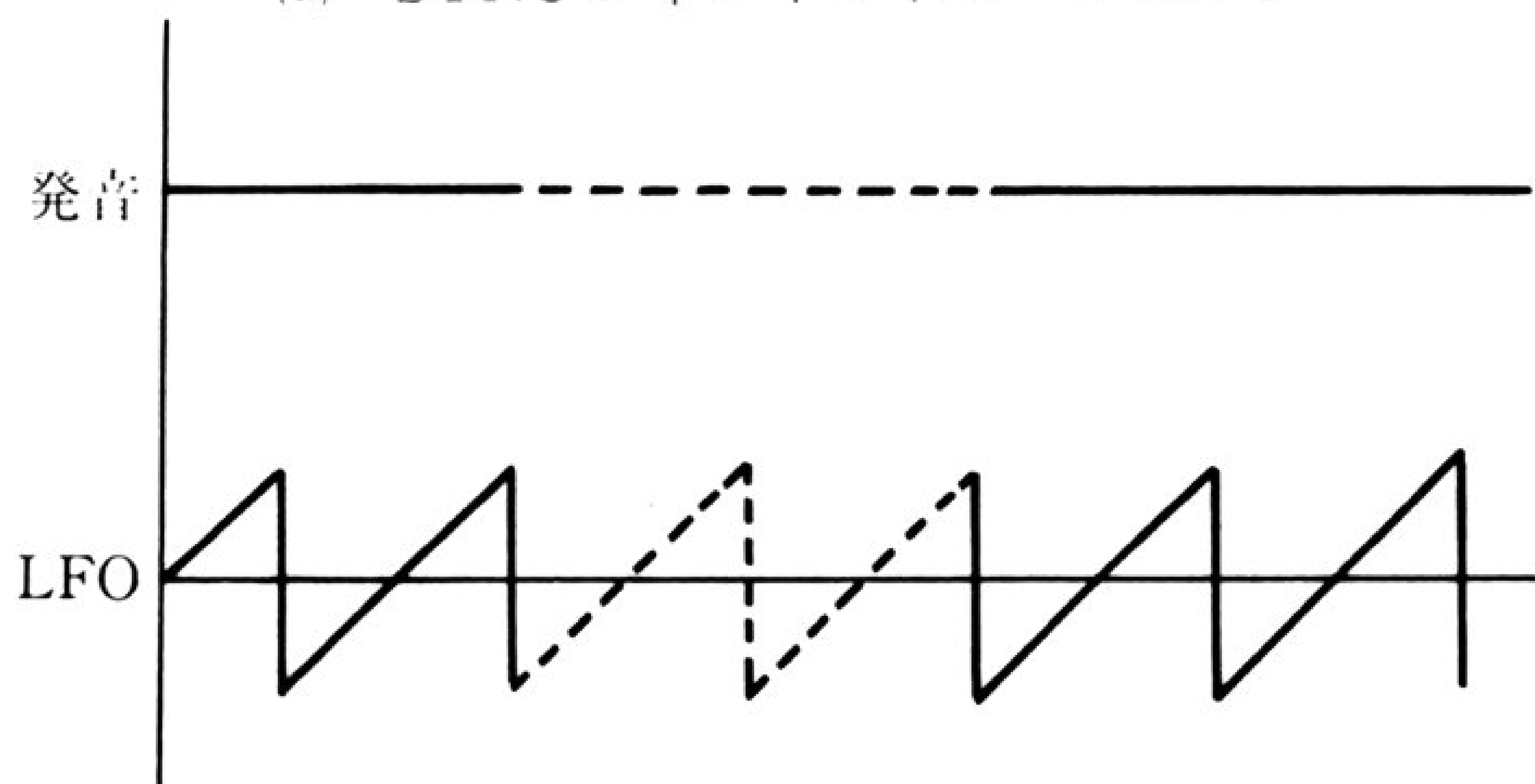


5：三角波ワンショット

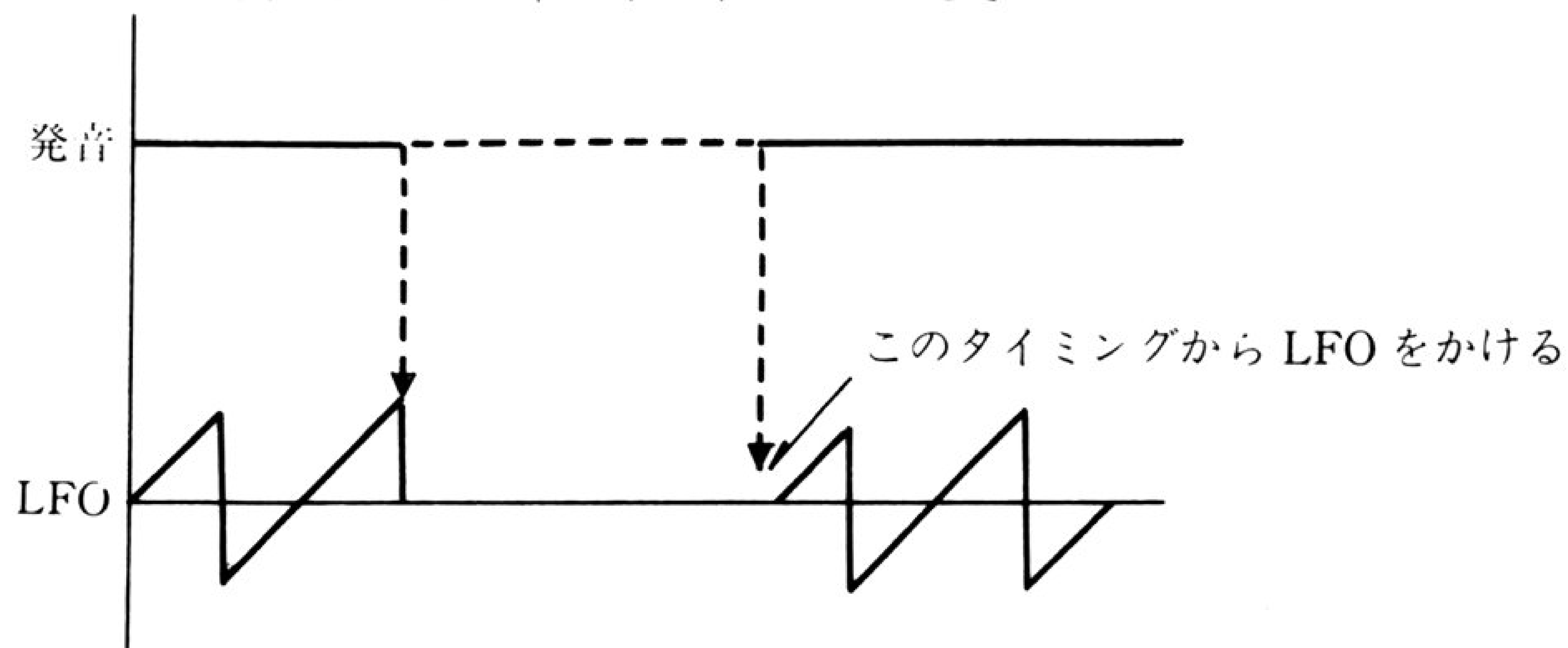
<SYNCディレイタイム>は0～255の値で，音が出てからLFO効果があらわれるまでの時間を与えます．

0のときは音の出るタイミングと非同期にLFO効果を付加します．1～255のときは，音が出てから与えられた時間だけ経過した後に同期してLFO効果を付加します．

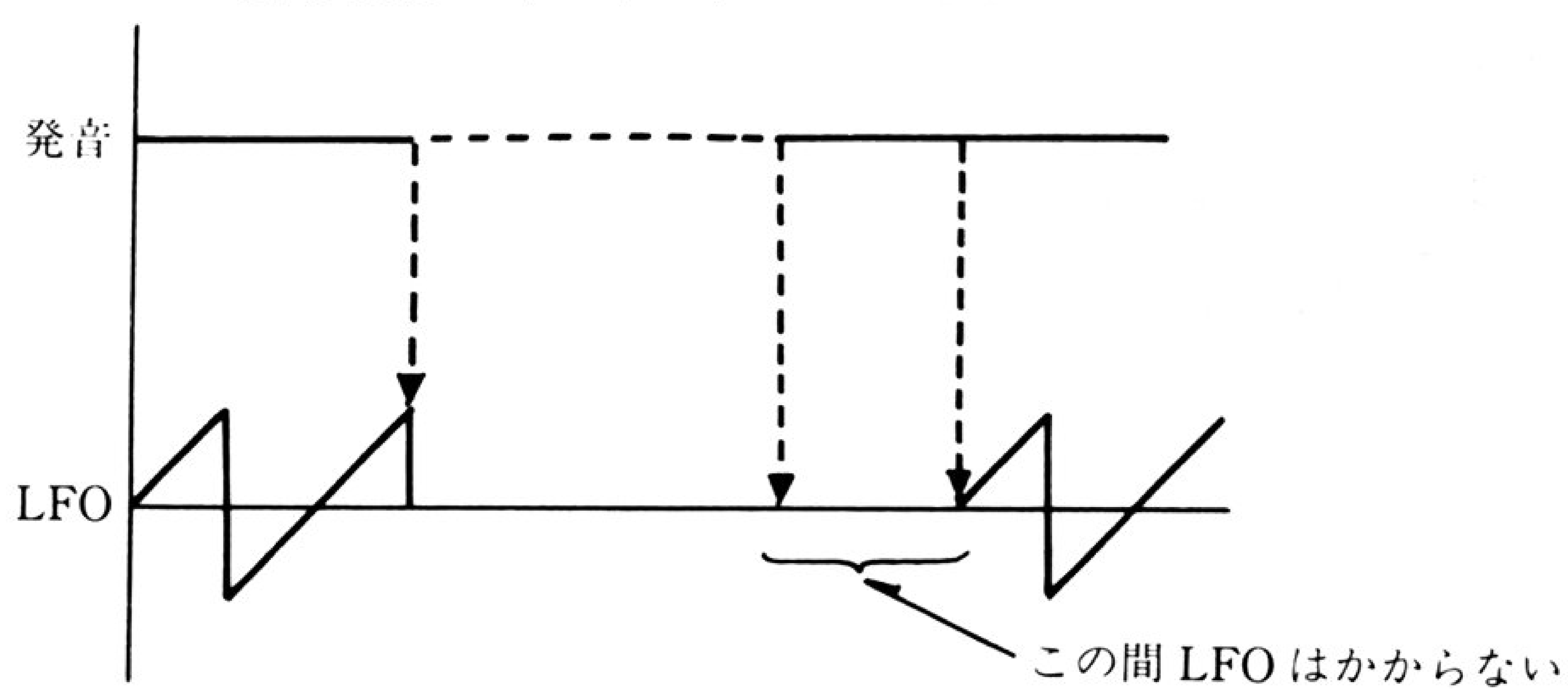
(a) SYNCディレイタイム＝0のとき



(b) SYNCディレイタイム＝1のとき



(c) SYNCディレイタイム＞1のとき



＜速さ＞は，0～16383の値でLFO波形の速度を与えます．

＜ピッチ変調深さ（微調整）＞は－127～127の値でビブラート効果のかかる深さを与えます．値が負の時はLFO波形が反転します．

＜振幅変調深さ（微調整）＞は－127～127の値でトレモロ効果のかかる深さを与えます．値が負の時はLFOの波形は反転します．

＜ピッチ変調深さ（粗調整）＞は0～15の値でビブラート効果のかかる深さをおおまかに設定します．

注　意　(1)　＜波形＞に4，5のワンショット波形を選んだ場合，必ず＜SYNCディレイタイム＞を1以上の値にして下さい．
1未満の値が指定されると出力される音程や音量が不定となります．

(2)　＜振幅変調深さ（微調整）＞のパラメータはFM音源にのみ有効です．

(3)　LFO効果はFM音源とSSG音源とでその効果が異なります．SSG音源ではFM音源に比べてピッチ変調が浅くかかるため，聴感上はLFO効果が付加されていない様に聞こえる場合があります．SSG音源の場合，FM音源よりも＜ピッチ変調深さ（粗調整）＞，＜ピッチ変調深さ（微調整）＞の値を大きめに設定して下さい．

VOICE REG

機　能	シンセサイザLSIのレジスタに値を設定します．
書　式	VOICE REG<レジスタ番号>，<式>
文　例	VOICE REG &H28,240

解　説　<レジスタ番号>は0～178の範囲で，書込むべきシンセサイザLSIの内部レジスタ番号を指定します．

<式>は0～255の範囲で書込む値を与えます．この命令はMMLのYr,dコマンドと同じ機能を行うものです．

注　意　**シンセサイザLSIの内部レジスタには拡張サウンド制御命令の制御に直接関係するものもありますので，レジスタを直接変更するような命令の実行には注意して下さい．**

特に，レジスタ番号&H21～&H27，&H2D～&H2Fの各レジスタへの書込みは行わないで下さい．

サンプルプログラム

```
10 '******** VOICE REG sample program ********
20 CLEAR ,&H9D00
30 PLAY ALLOC 255
40 PLAY "@65@V120C"
50 VOICE REG &H28,&HF0:FOR I=0 TO 5000:NEXT
60 VOICE REG &H28,&H0
70 END
```

VOICE INIT

機　能　**音色バンクを初期化します.**

書　式　VOICE INIT

文　例　VOICE INIT

解　説　音色バンクを初期状態にもどします.
FM音源の音色番号は0（デフォルト）に設定されます．ただし，SSG音源は初期化されません．
VOICE命令で使用した配列はこの命令により拡張サウンドインタプリタより解放されます．

ON PLAY (c,n) GOSUB

機 能　PLAY割込処理ルーチンの開始行を定義します.

書 式　ON PLAY（<チャンネル番号>，<残りバイト数>）
GOSUB<行番号またはラベル>

文 例　ON PLAY（1，1024）GOSUB ＊NEXT.PLAY

解 説　バックグラウンドで演奏中，<チャンネル番号>のサウンドバッファ内の未演奏音楽情報が<残りバイト数>以下になった時に，分岐する処理ルーチンの開始行を定義します.

<チャンネル番号>が0のときは全チャンネルの中で最大の残りバイト数を持つものについて，また<チャンネル番号>が負のときは，最小の残りバイト数を持つものについて割込を発生させます.

<残りバイト数>は，0～32767の範囲です.

サンプルプログラム

```
10 '********** PLAY INTERRUPT sample program **********
20 CLEAR ,&H9EB0:I=0
30 GOSUB *BEGIN:GOSUB *PLAYSUB
40 ON PLAY (3,0) GOSUB *PLAYSUB
50 PLAY ON
60  X=RND*639:Y=RND*399:C=(RND*100 MOD 15)+1
70  CIRCLE (X,Y),RND*30,C,,,,F,(RND*100 MOD 15)+1
80 GOTO 50
90  REM
100 *PLAYSUB
110 PLAY STOP:CLS 3
120 PLAY A$,B$,B$
130 PLAY B$,A$,B$
140 PLAY B$,B$,A$
150 PLAY ON
160 RETURN
170 *BEGIN
180 CONSOLE 0,25,0,1
190 SCREEN 3,0 : CLS 3 : COLOR ,,,,2
200 PLAY ALLOC 255,255,255
210 INI$="MB@3"
220 PLAY INI$,INI$,INI$
230 A$="T165@55@V100O4L8C4CC4DE4EE4ED4CD4EC4CO3G4RO4E4.EEFG4GG4GF4EF4GE4.R4GE2RG
E2RGER8GER8GE4.R4"
240 B$="R2R2R2R2R2R2"
250 I=I+1
260 RETURN
```

PLAY　ON/OFF/STOP

機　能 PLAY割込の許可（ON）/禁止（OFF）/停止（STOP）を制御します．

書　式 PLAY ON/PLAY OFF/PLAY STOP

文　例 PLAY ON

解　説 PLAY ON：PLAY割込を許可します．以降，割込み条件が満たされると割込み処理ルーチンに分岐します．

PLAY OFF：PLAY割込を禁止します．以降，割込み条件が満たされても処理ルーチンには分岐しません．

PLAY STOP：PLAY割込を停止します．以降，割込条件が満たされても処理ルーチンには分岐しません．しかし，割込のあったことは記憶しており，後でPLAY ONによって割込みが許可された時に処理ルーチンに分岐します．

注　意 PLAY文の割込はN_{88}-BASICの他の割込命令と異なり，割込処理ルーチンに制御が移っても自動的に割込停止状態とはなりません．したがって，処理ルーチンの先頭ではPLAY OFFまたはPLAY STOPを実行し，処理ルーチンからリターンする直前にPLAY ONを実行する必要があります．

STATUS PLAY (C)

機　能　**サウンドバッファの未演奏データのバイト数を返します．**

書　式　STATUS PLAY (〈チャネル番号〉)

文　例　PRINT STATUS PLAY(0)

解　説　〈チャネル番号〉で示されるチャネルのサウンドバッファの未演奏データのバイト数を与えます．

〈チャネル番号〉が１～６の時は，対応するチャネルの未演奏データバイト数を，０のときには最大のバイト数を与えます．また，負のときには最小のバイト数を与えます．

サンプルプログラム

```
10 '********** STATUS PLAY sample program **********
20 CLS
30 CLEAR ,&H9D00
40 PLAY ALLOC 255,255
50 PLAY "MB@66@V100L2CDEFGAB>C","MB@66@V100L2EFGAB>CDE"
60 LOCATE 15,10
70 PRINT "REST BYTE (CH-1)=";
80 PRINT STATUS PLAY (1),
90 PRINT "REST BYTE (CH-2)=";
100 PRINT STATUS PLAY (2)
110 IF STATUS PLAY (-1)=0 THEN 130
120 GOTO 60
130 END
```

19.2.5　他の命令との関係

(1)　CLEAR文，END文

バックグラウンドで演奏が行われている時に，CLEAR文あるいはEND文が実行されても演奏は中断されませんが，サウンドバッファ領域は初期化されますので，その後サウンド関連命令を使用する時にはPLAY ALLOC文を実行し直す必要があります．

(2)　STOP文/STOPキー

バックグラウンドでの演奏はいったん停止します．その後，バックグラウンドでの演奏に影響を与えるような命令が実行されず，かつ再実行可能であればバックグラウンドでの演奏もCONTコマンドで再開されます．また，バックグラウンドでの演奏を一旦停止させている時に，ダイレクトモードでPLAY文を実行すると，サウンドバッファに新しいデータを格納（つまりPLAY文を解釈）して演奏が再開されます．このデータを初期化するためには，PLAY ALLOC文を実行して下さい．

(3)　サウンドバッファの取扱いに関して

バックグラウンドによる演奏が行われている際に，BLOAD文あるいはPOKE文によりサウンドバッファが乱されますと正しい演奏が行えませんので御注意ください．

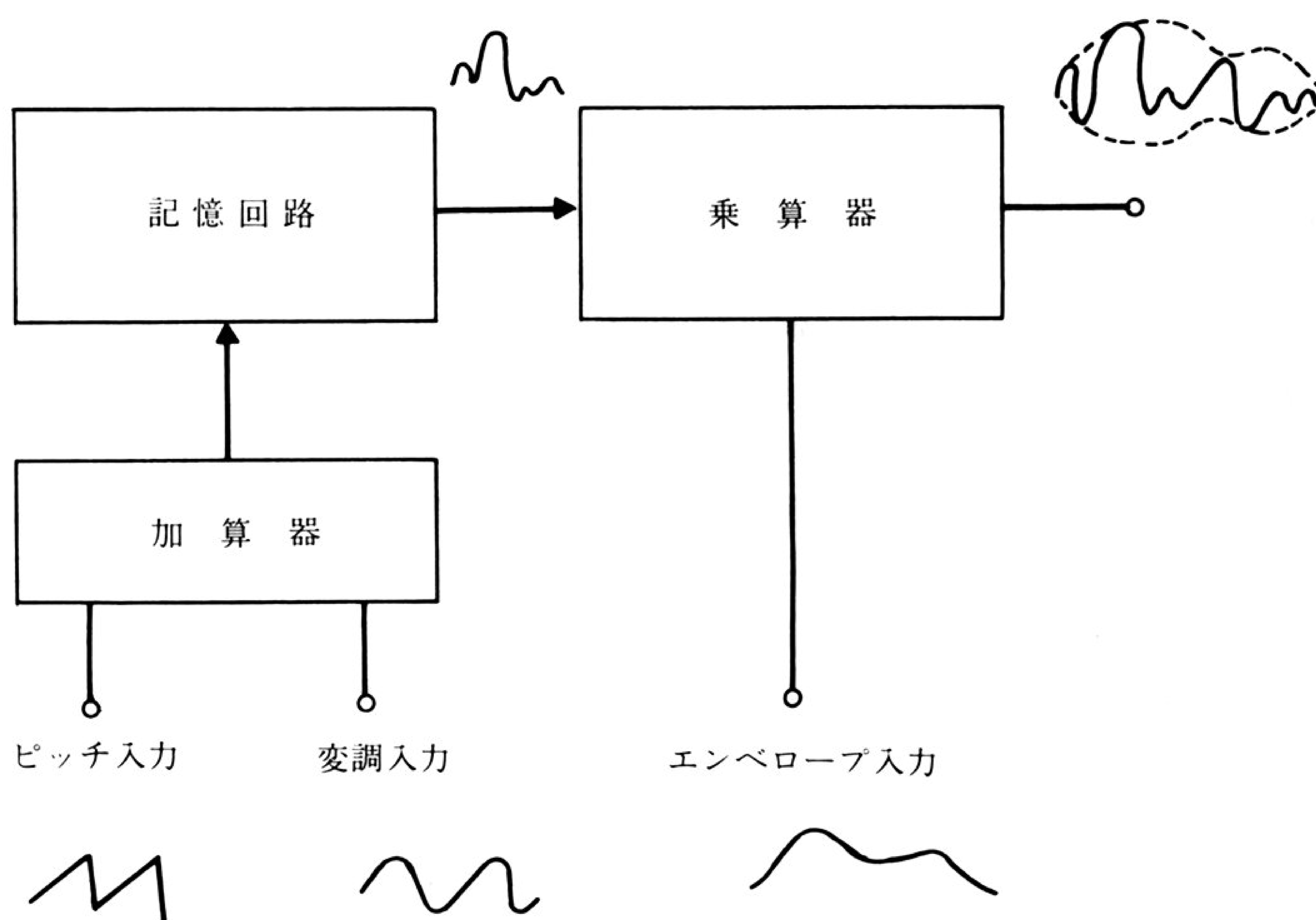
(4)　エラーが起きた時

現在設定中の音色が初期化され演奏が中断されます．

19.2.6　より良い演奏のために

(1)　FM音源について

サウンドボードに内蔵されているシンセサイザLSIのチャネル１～３はFM方式の音源と



なっています．ここでは，FM方式の音源についてその原理を簡単に説明します．

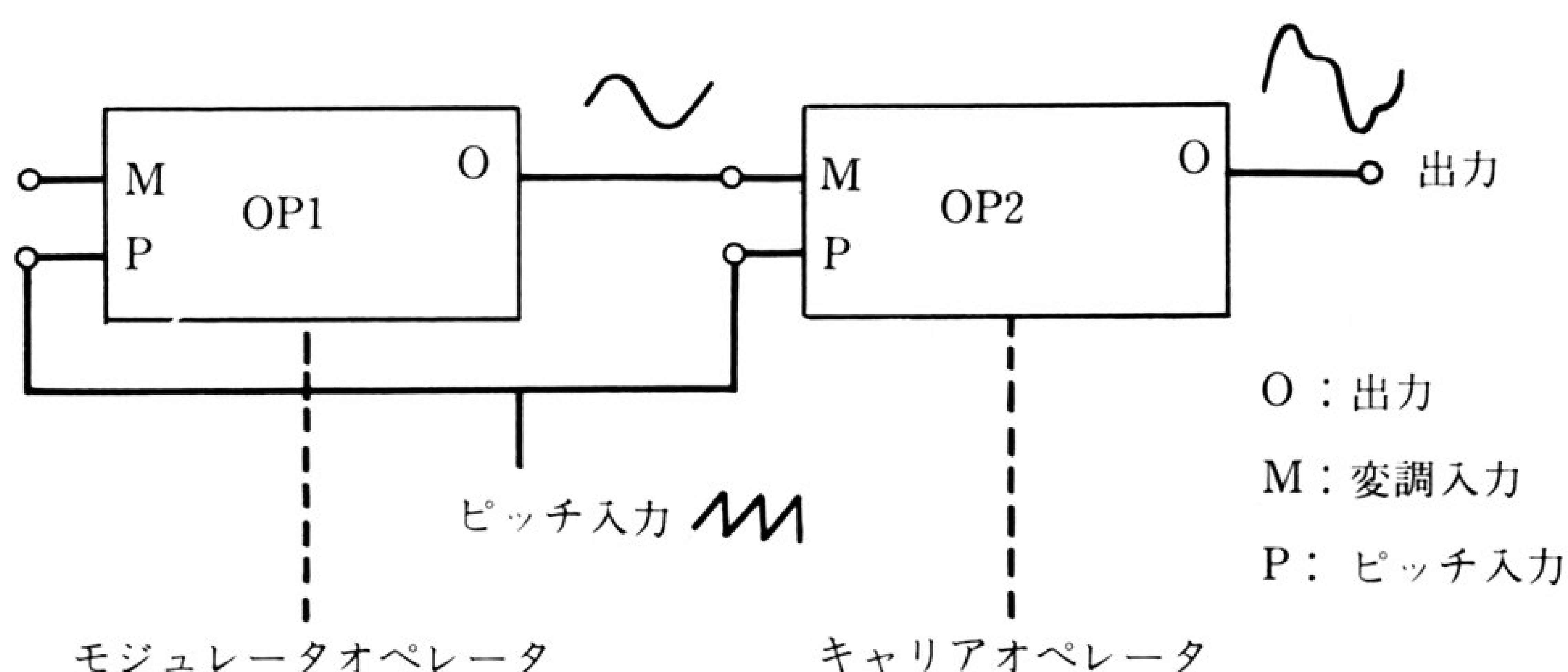
この図は，FM音源のオペレータ（Operator）を簡略化して表したものです．記憶回路には，正弦波の波形が記憶されており，ピッチ入力の読出し波形（ノコギリ波）により記憶回路から波形が再生されます．この，読出波形の周波数が出力される波形の周波数を決定します．

波形を読出す時に，ノコギリ波と同時に別な波形（例えば正弦波）を加えると記憶回路からの出力は，別に加えられた波形（変調入力）の周波数や振幅，波形によって大きく異なる複雑なものとなります．

ピッチ入力と変調入力により造られた波形は，エンベロープ入力からの振幅波形とかけ合わされ，最終的な出力として得られます．

このようにして，FM音源は複雑な音色を発生させます．

サウンドボードは，FM音源の各チャネル毎に4つのオペレータを持っています．第1オペレータは，自分自身の出力を変調信号にしていますが，それ以外のオペレータはピッチ入力や変調入力に対して，どの出力をどう組み合わせるかを選択することができます．この組み合わせをアルゴリズム（Algorithm）と呼びます．

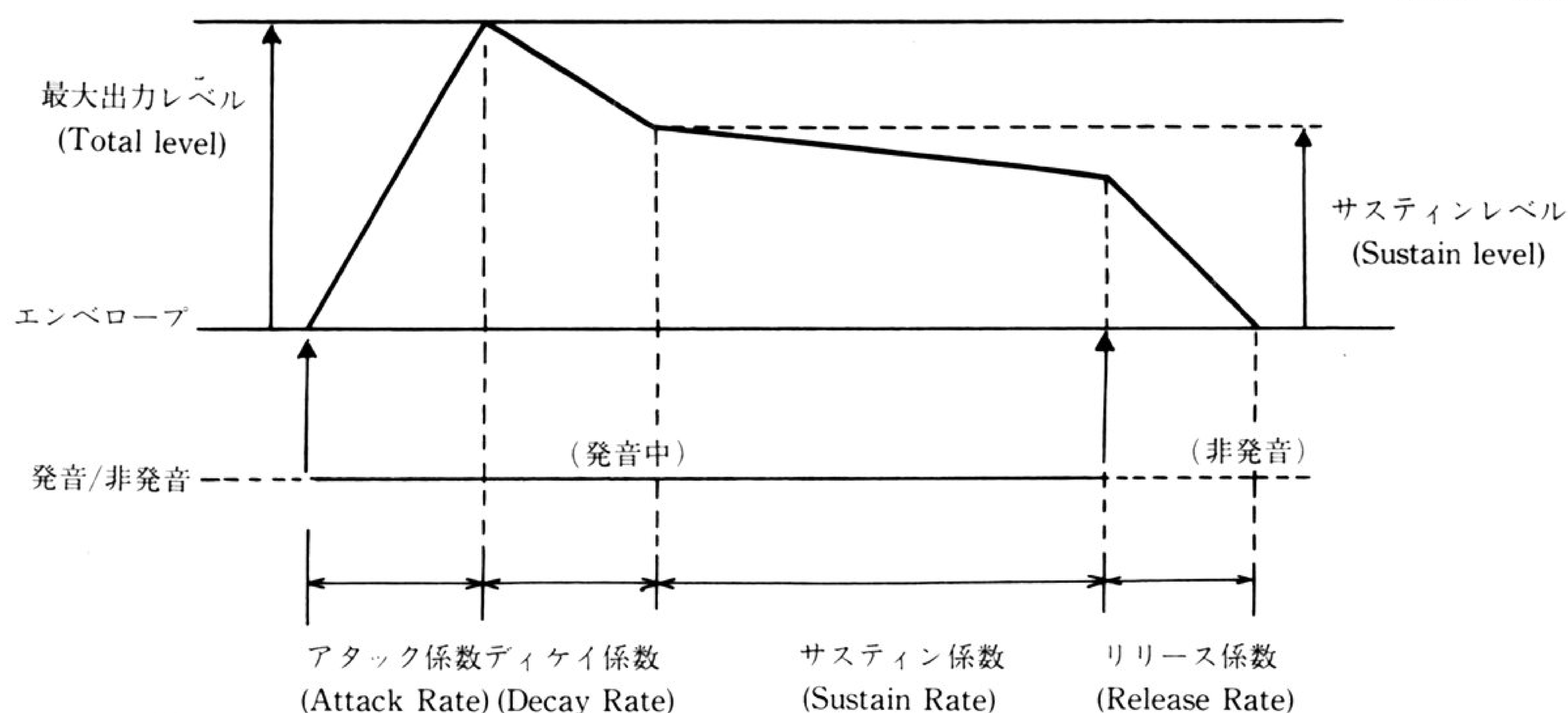


今，このような2つのオペレータからなるアルゴリズムを仮定します（エンベロープ入力は省略）．OP1の出力は，OP2の変調入力に加えられています．OP1はOP2のための変調波形をつくっているわけで，このような使い方でのOP1のことを“モジュレータオペレータ”，OP2を“キャリアオペレータ”と呼びます．

(2) エンベロープジェネレータ(EG) について

FM音源からの出力に音量の時間的な変化を与えるためにエンベロープ入力に波形を加えます．この波形の発生を行うのがエンベロープジェネレータ（Envelop Generator：EG）です．

エンベロープジェネレータは次のようなパラメータを持っており，音に様々な表情をつけ加えることができます．



アタック係数（Attack Rate：AR）

発音してからEGの出力レベルが最大になるまでの時間を定めます．

0が最短で31が最長です．

ディケイ係数（Decay Rate：DR）

EGの出力レベルが最大からサスティンレベルに達するまでの時間を定めます．

0が最短で31が最長です．

サスティン係数（Sustain Rate：SR）

サスティンレベルからEGの出力レベルが0になるまでの時間を定めます．

0が最短で31が最長です．

リリース係数（Release Rate：RR）

発音が終わってからEGの出力レベルが0になるまでの時間を定めます．

0が最短で15が最長です．

最大出力レベル（Total Level：TL）

EGの最大出力レベルを定めます．0が最小で127が最大です．

サスティンレベル（Sustain Level：SL）

Decay状態からSustain状態に移るときのレベルをを定めます．

0が最小で15が最大です．

(3)　音色パラメータについて

拡張サウンド制御命令では，FM音源での音創りを容易に行うための命令を用意しています．利用者は音色データを，直接ハードウエアに設定するような複雑な手順を経ずに容易に音色を変更したり創り出したりすることができます．この音色創りのために「音色パラメータ」と呼ばれる値をセットする必要があります．

サウンドボードはチャネル1～6について独立したパラメータ領域を各々持っており，MML@xコマンドが実行される度に音色パラメータデータをその領域に移します．MMLのZコマンドやVOICE LFO文はこのパラメータ領域に値を直接設定するものです．また，VOICE文はこのパラメータ領域に与えるための音色パラメータデータを予め用意するものです．では，以下に音色パラメータの各々について説明していきます．なお，LFO関係のパラメータについては，先にVOICE LFO文の説明で，またエンベロープ関係のパラメータについては既に述べましたので簡単に説明します．

音色パラメーター覧

<table>
<tr><th>音色パラメータ番号</th><th>音色用配列要素</th><th>パラメータ名
：略称名</th><th>説　　明</th></tr>
<tr>
<td>0</td>
<td>(0，0)</td>
<td>フィードバック/アルゴリズム
(Feed Back/Algorithm)
：FB/ALG</td>
<td>
6ビットからなる値です
MSB　LSB
X X F2 F1 F0 A2 A1 A0
アルゴリズム
フィードバック

○フィードバック：F_2～F_0　0～7
第1オペレータは自分の出力を変調入力としていますが，その変調度を決定します．
<table>
<tr><td>F</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr>
<tr><td>変調度</td><td>OFF</td><td>π/16</td><td>π/8</td><td>π/4</td><td>π/2</td></tr>
</table>
<table>
<tr><td>5</td><td>6</td><td>7</td></tr>
<tr><td>π</td><td>2π</td><td>4π</td></tr>
</table>
○アルゴリズム：A_2～A_0　0～7
4つのオペレータのアルゴリズムを決定します．
<table>
<tr><th>A</th><th>アルゴリズム</th></tr>
<tr><td>0</td><td>P OP1 OP2 P OP3 OP4</td></tr>
<tr><td>1</td><td>P OP1 OP3 OP4 OP2 P</td></tr>
<tr><td>2</td><td>P OP2 OP3 OP4 OP1 P</td></tr>
<tr><td>3</td><td>P OP1 OP2 OP4 P OP3 P</td></tr>
</table>
</td>
</tr>
</table>

音色パラメータ番号	音色用配列要素	パラメータ名 ：略称名	説　　明
			<table><tr><td>4</td><td>P OP1 OP2 P P OP3 OP4 P</td></tr><tr><td>5</td><td>P OP1 OP2 OP3 OP4</td></tr><tr><td>6</td><td>P OP1 OP2 P OP3 P OP4</td></tr><tr><td>7</td><td>P OP1 P OP2 P OP3 P OP4</td></tr></table>Pはピッチ入力を示します.

音色パラメータ番号	音色用配列要素	パラメータ名 ：略称名	説明
5	（0，1）	オペレータマスク (Operator Mask) ：OPMSK	4ビットからなる値です． 0～15 X X X X O_4 O_3 O_2 O_1 ○オペレータON/OFF：O_4～O_1 各オペレータの使用/不使用を与えます． O_4～O_1の各ビットが“1”のとき使用 “0”のとき不使用となります．
10	（0，2）	LFO変調波形 (LFO Wave Form) ：WF	LFOの変調波形を決定します． 0～5 0：ノコギリ波形 1：矩形波 2：三角波 3：サンプル＆ホールド 4：ノコギリ波ワンショット 5：三角波ワンショット (→VOICE LFO文)
15	（0，3）	LFO SYNCディレイ (LFO Sync Delay) ：SYNC	LFO変調のSyncディレイタイムを与えます． 0～255 (→VOICE LFO文)
20	（0，4）	LFO速さ (LFO Speed) ：SPEED	LFO変調のかかる速さを与えます． 0～16383 (→VOICE LFO文)
25	（0，5）	LFOピッチ変調深さ (LFO Pitch Depth) ：PD	LFOピッチ変調のかかる深さを細かく調整します．　－127～127 (→VOICE LFO文)
30	（0，6）	LFO振幅変調深さ (LFO Amp Depth) ：AD	LFO振幅変調のかかる深さを細かく調整します．　－127～127 (→VOICE LFO文)

音色パラメータ番号	音色用配列要素	パラメータ名：略称名	説　　明
35	(0, 7)	LFOピッチ変調深さ (LFO Pitch Sense) ：PS	LFOピッチ変調のかかる深さを大まかに調整します.　0～15 (→VOICE LFO文)
40 45	(0, 8) (0, 9)	未使用	(このパラメータは使用されません)
1 ≀ 4	(1, 0) (2, 0) (3, 0) (4, 0)	アタック係数 (Attack Rate) ：AR	各オペレータのアタック係数を与えます. 0～31 (→(2)「エンベロープジェネレータ(E.G.)について」)
6 ≀ 9	(1, 1) (2, 1) (3, 1) (4, 1)	ディケイ係数 (Decay Rate) ：DR	各オペレータのディケイ係数を与えます. 0～31 (→(2)「エンベロープジェネレータ(E.G.)について」)
11 ≀ 14	(1, 2) (2, 2) (3, 2) (4, 2)	サスティン係数 (Sustain Rate) ：SR	各オペレータのサスティン係数を与えます. 0～31 (→(2)「エンベロープジェネレータ(E.G.)について」)
16 ≀ 19	(1, 3) (2, 3) (3, 3) (4, 3)	リリース係数 (Release Rate) ：RR	各オペレータのリリース係数を与えます. 0～15 (→(2)「エンベロープジェネレータ(E.G.)について」)

音色パラメータ番号	音色用配列要素	パラメータ名 ：略称名	説　　明
21 ≀ 24	(1, 4) (2, 4) (3, 4) (4, 4)	サスティンレベル (Sustain Level) ：SL	各オペレータのサスティンレベルを与えます. 0～15 (→(2)「エンベロープ・ジェネレータ(E.G.)について」)
26 ≀ 29	(1, 5) (2, 5) (3, 5) (4, 5)	最大出力レベル (Total Level) ：TL	各オペレータの出力レベルを与えます. 0～127 (→(2)「エンベロープ・ジェネレータ(E.G.)について」)
31 ≀ 34	(1, 6) (2, 6) (3, 6) (4, 6)	キーボードレイトスケーリング　深さ (Key-board Rate Scaling Depth) ：KS	各オペレータのキーボードレイトスケーリング深さを与えます. このパラメータは音に表情を与えるため，高い音になる程エンベロープの各Rateを短くします. 0で最小（高い音も低い音も同じエンベロープ）3で最大です.　0～3
36 ≀ 39	(1, 7) (2, 7) (3, 7) (4, 7)	マルチプル (Multiple) ：ML	各オペレータに対して与える周波数の比を与えます. FM音源の各オペレータには，基本となる周波数の整数比倍の値を与えることができます. 0が1/2倍 15が15倍の周波数をオペレータに与えます. 0～15
41 ≀ 44	(1, 8) (2, 8) (3, 8) (4, 8)	デチューンレイト (Detune Rate) ：DT	各オペレータのデチューンレイトを与えます. このパラメータは，コーラス効果や微妙なゆらぎ効果を出すために，各オペレータの周波数をわずかにずらします. －4～3

音色パラメータ番号	音色用配列要素	パラメータ名 ：略称名	説　　　明
46 ～ 49	(1, 9) (2, 9) (3, 9) (4, 9)	LFO振幅変調深さ (LFO Amp. Sense) ：AS	各オペレータのLFO振幅変調深さを，大まかに与えます． 0～15 (→VOICE LFO文)

以上のパラメータは，VOICE LFO文の時は有効範囲のチェックを行いますが，VOICE文やMMLのZコマンドでは与えられたパラメータではチェックされません．そのために，思った様な音が出ない事がありますので注意して下さい．

音色パラメータはシンセサイザLSIの内部レジスタと密接に関係しており，LFO関連のパラメータ以外は対応するレジスタが必ず一つは存在します．例えば，パラメータML（マルチプル）とDT（デチューンレイト）は，内部レジスタの&H30～&H3E番の下位4bitと上位3bitに対応します．VOICE REG文やMMLのYコマンドで直接内部レジスタを操作する場合には，次の「FM/SSG音源内部レジスタ一覧」を参考にして下さい．

注　意　(1)　OUT命令によりシンセサイザLSIのレジスタを操作した場合，拡張サウンド制御命令の動作は保証しません．

(2)　VOICE REG文及びYコマンドにより次のレジスタを操作しないでください．

レジスタ番号　&H21～&H27，&H2D～&H2F

もし，これらのレジスタ値を書いたり読んだりした場合，サウンド制御命令の動作は保証されません．

(3)　SSG音源に対するパラメータはLFO関係のパラメータのうちピッチ変調に関するものだけ有効です．

FM/SSG音源内部レジスタマップ

レジスタ番号	レジスタマップ D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	対応チャネル	説明	備考
&H21	TEST	X	テスト用レジスタ	ユーザアクセス禁止
&H24	TIMER-A(Ⅰ)	X	タイマAレジスタ（上位）	ユーザアクセス禁止
&H25	TIMER-A(Ⅱ)	X	タイマAレジスタ（下位）	ユーザアクセス禁止
&H26	TIMER-B	X	タイマBレジスタ	ユーザアクセス禁止
&H27	MODE / RESET B A / ENABL B A / LOAD B A	X (CH 3)	CH 3モード及びタイマ制御	ユーザアクセス禁止
&H28	OPERATOR 4 3 2 1 / CH	1～3	発音/非発音	
&H30 &H31 &H32	DT_1 / ML_1	1 2 3	オペレータ 1 デチューン/マルチプライ	
&H34 &H35 &H36	DT_3 / ML_3	1 2 3	オペレータ 3 デチューン/マルチプライ	
&H38 &H39 &H3A	DT_2 / ML_2	1 2 3	オペレータ 2 デチューン/マルチプライ	
&H3C &H3D &H3E	DT_4 / ML_4	1 2 3	オペレータ 1 デチューン/マルチプライ （以下，各レジスタの配置は同様）	
&H40 | &H4E	TL 1～4	1～3	最大出力レベル	
&H50 | &H5E	KS 1～4 / AR 1～4	1～3	キーボードレートスケーリング深さ アタック係数	
&H60 | &H6E	DR 1～4	1～3	ディケイ係数	
&H70 | &H7E	SR 1～4	1～3	サスティン係数	
&H80 | &H8E	SL 1～4 / RR 1～4	1～3	サスティンレベル リリース係数	
&HA0 &HA1 &HA2	F-number 1	1 2 3	F-number 下位	
&HA4 | &HA6	Block / F-number 2	Block 1～3	F-number 上位 （&HA0～&HA2と同様）	
&HA8 &HA9 &HAA	3 ch F-number 1	3	MODE 2，3の時の CH 3のオペレータ F-number 下位	オペレータ 1 オペレータ 2 オペレータ 3
&HAC | &HAE	3 ch Block / 3 CH F-number2	3	MODE 2，3の時の CH 3のオペレータ Block/F-number 上位	
&HB0 &HB1 &HB2	FB / ALG	1 2 3	フィードバック/アルゴリズム	

（注）　内部レジスタの設定値のうち，TL,AR,DR,SR,SL,RRは
音色パラメータとは大小/長短関係が逆になっています．

例）　ＴＬ：０～127　　→レジスタ設定値
（大）（小）
ＴＬ：０～127　　→音色パラメータ
（小）（大）

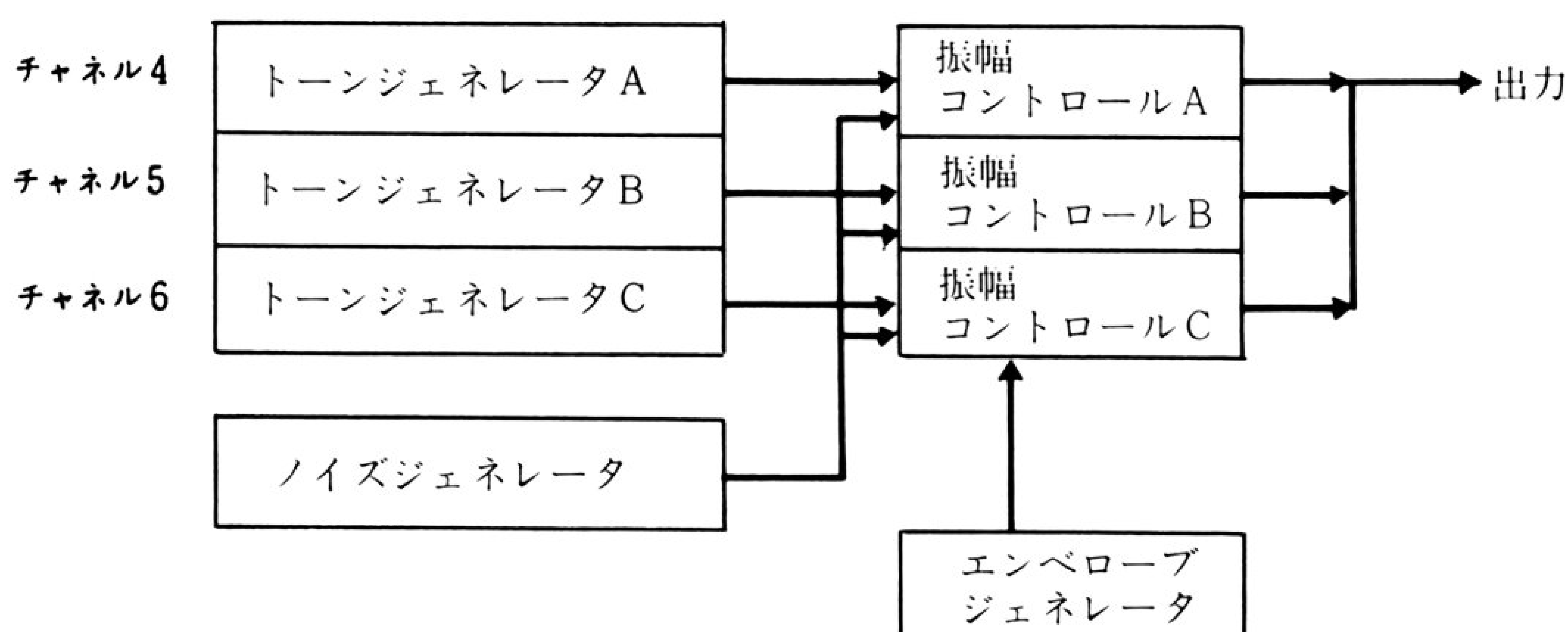
レジスタ番号	レジスタマップ D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0				対応チャネル	説　　明	備　　考
&H00	TUNE11				4	CH 4　周波数レジスタ下位	
&H01		TUNE12			4	CH 4　周波数レジスタ上位	
&H02	TUNE21				5	CH 5　周波数レジスタ下位	
&H03		TUNE22			5	CH 5　周波数レジスタ上位	
&H04	TUNE31				6	CH 6　周波数レジスタ下位	
&H05		TUNE32			6	CH 6　周波数レジスタ上位	
&H06		NOISE			4～6	ランダムノイズ周波数	
&H07	"1" "0"	NOISE MIX	TONE MIX		4～6	ノイズ/楽音ミキサーコントロール	上位2 bit 変更禁止
&H08		F/V	Level		4	固定量/可変音量コントロール 出力音量	
&H09					5		
&H0A					6		
&H0B	ENVELOP PER.₁				4～6	エンベロープ周期　下位	
&H0C	ENVELOP PER.₂				4～6	エンベロープ周期　上位	
&H0D		SHAPE			4～6	エンベロープ形状	

（注）　SSG/FM用レジスタのうち未定義のものやユーザアクセスが禁止されているものをアクセスした場合の動作は保証しません.

(4) SSG音源について

チャネル4～6はSSG音源となっています．

SSG音源部の構成は次のようになっています．



○トーンジェネレータ：〔レジスタNo. &H00～&H05〕

デューティ比1：1の方形波を発生する発振器です．

各チャネルの発振周波数は次式により求められます．

Fnote＝667×1000/(32×To)

（Fnote：発振周波数
To　：トーンジェネレータ設定値（0～4095））

○ノイズジェネレータ：〔レジスタNo &H06〕

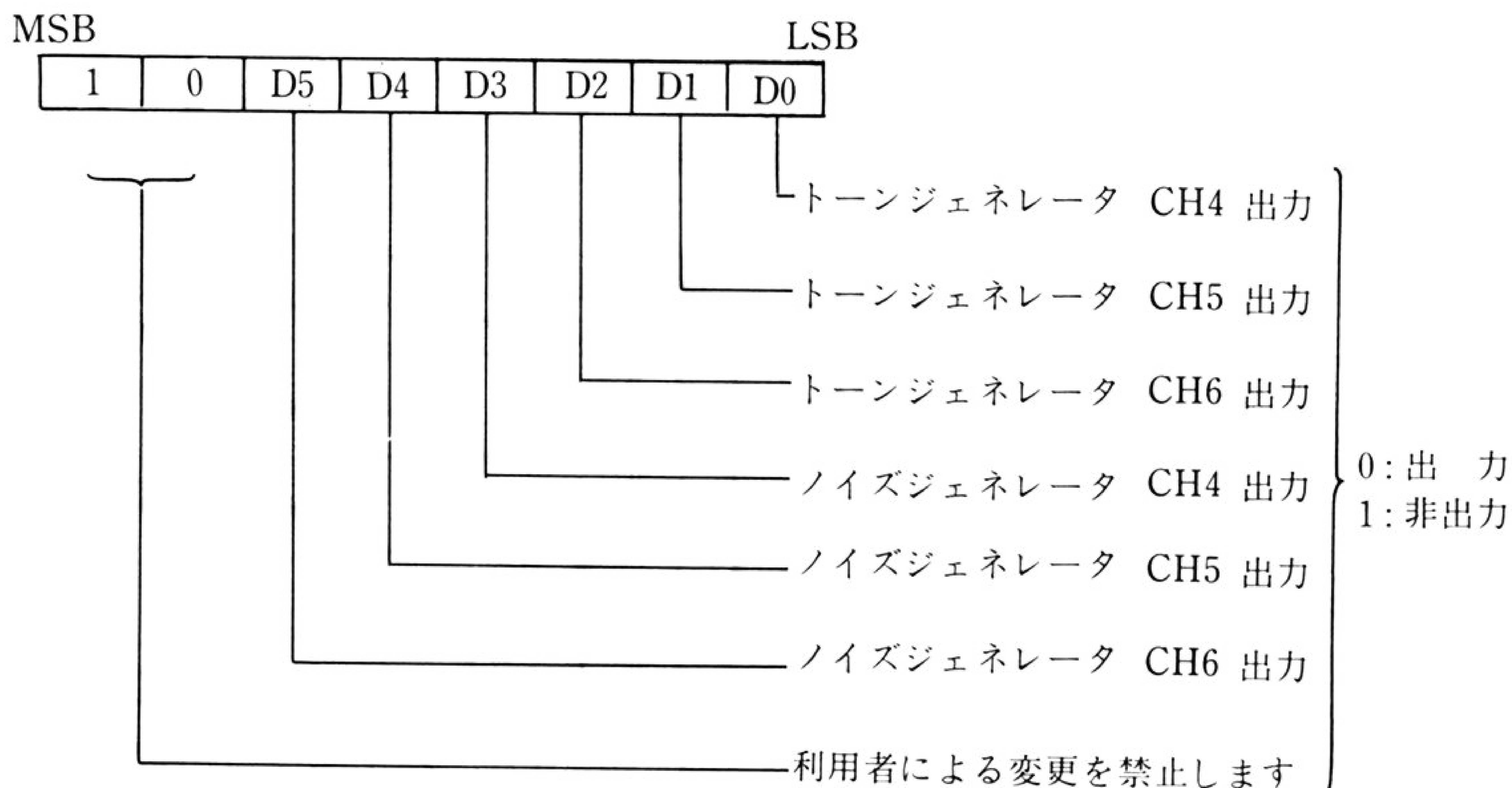
類似ランダムノイズを発生します．レジスタ&H06に設定される値によりノイズ周波数を可変できます．

Fnois＝667×1000/(32×Tn)

（Fnois：ノイズ周波数（乱数系列発生周期）
Tn　：ノイズジェネレータ設定値（0～31））

○ミキサーコントロール〔レジスタNo. &H07〕

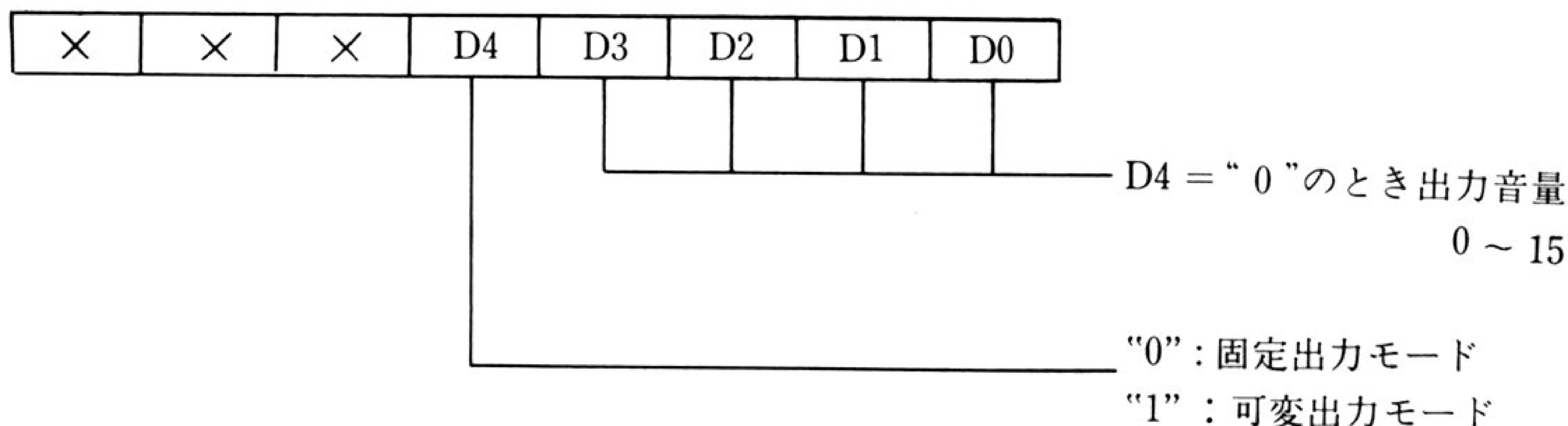
3つのトーンジェネレータとノイズジェネレータを出力チャネルに出すか否かを設定します．このレジスタは次のようなビット構成になっています．



注 意 レジスタNo.&H07のビット6，7は"10"以外を設定してはいけません．"10"以外の設定を行った場合，動作は保証されません．

○振幅コントロール〔レジスタNo. &H08～&H0A〕

各チャネルの出力音量，エンベロープジェネレータによる可変出力を制御します．SSG音源に対するMMLのVコマンドはこのレジスタへの設定に他なりません．



○エンベロープジェネレータコントロール〔レジスタNo. &H0B～&H0D〕

変化のあるエンベロープをつくり出すために2つの値を設定します．ひとつは，レジスタ&H0B，&H0Cに設定するエンベロープ周期で，これはMMLのMxコマンドに対応します．エンベロープ周波数は次式で求められます．

$$Fenv=667\times1000/(512\times Te)$$

(Fenv：エンベロープ周波数（Hz）
Te　：エンベロープ周期設定値　0～65535)

もうひとつは，レジスタ&H0Dに設定するエンベロープ形状でこれはMMLのSxコマンドに対応します．

レジスタ&H0Dとエンベロープの形状の関係は次のとおりです．

設定値	対応するエンベロープ
0 ～ 3	
4 ～ 7	
8	
9	
10	
11	
12	
13	
14	
15	

注意　タイマーレジスタ（レジスタ番号&H24～&H27）は拡張サウンド制御命令が内部で使用しますので書込は禁止します．これらのレジスタに書込を行った場合の動作は保証しません．

(5) FM音源のモードについて

FMチャネルのうちCH 3は以下のような3つのモードを持ちます．

楽音モード：通常の音楽演奏に適したモードです．

発生周波数は内部レジスタ&HA2，&HA6で指定します．

効果音モード：効果音の発生に適する様に各オペレータは独立して発生周波数を設定できます．

第1オペレータ：&HA9，&HAD

第2オペレータ：&HAA，&HAE

第3オペレータ：&HA8，&HAC

第4オペレータ：&HA2，&HA6

CSMモード：発生周波数は効果音モードと同様です．

ただし，このモードの時には内部タイマーレジスタAのカウンタがオーバフローした時に音を発生します．

（発生周波数の求め方）

FMチャネルCH 1～3に与える周波数データ（Block/F-number）は次の式で計算します．発生する周波数と，その周波数のオクターブをもとにして，Block値とF-number値を計算します．

X	X	Block	F-number

Block＝オクターブデータ

F-number＝（Fvo× 2^{20}/55556)/$2^{Block-1}$

Fvo＝発生したい周波数（Hz）

（例）A 4 ＝440Hzの場合

Block＝ 4

F-number＝（440× 2^{20}/55556)/2^{4-1}＝1038

(6)　ミュージックジェネレータボード（PC-9801-14）用プログラムの移植について

拡張サウンド制御命令は，拡張ミュージック制御命令と高い互換性をもっています．既存のミュージックジェネレータ用のプログラムをサウンド用に移植することも，MML（ミュージックマクロランゲージ）単位では大きな変更なしに行うことができます．

○拡張サウンド制御命令は6声，拡張ミュージック制御命令は8声までの出力をサポートしていますので，ミュージック→サウンドの移植にあたっては，2CH分を削って下さい．

○PLAY文MMLでは次の様な変更が必要です．

（Sコマンド）

ミュージック制御命令では，

S〈アタック係数〉，〈ディケイ係数〉，〈サスティンレベル〉，〈リリース係数〉，〈サスティン係数〉

となっていますが，サウンド制御命令では，SコマンドはCH4～6にのみ有効で，書式は

S〈エンベロープ形状〉

となっています．CH1～3に対するエンベロープは音色を切換えるか利用者が音色を作成するかして下さい．なお，音色パラメータのエンベロープパラメータの数値とミュージック制御命令Sコマンドの数値には互換性はありません．

（Oコマンド）

ミュージック制御命令では，O4～O7まで可能ですが，サウンド制御命令では，O1～O8（O9）まで可能です．

（Kコマンド）

ミュージック制御命令では，“K1”～“K49”まで可能で，“O4C”に対応するものは“K1”でした．サウンド制御命令では“K0”～“K96”まで可能で“O4C”に対応するのは“K36”です．

（Tコマンド）

ミュージック制御命令は，各チャネル毎にテンポを設定できますが，サウンド制御命令では全チャネルを通じて1つです．

また，PLAY文の文字列では，記述されている文字列の最後のテンポ指定が有効です．

例えば，　PLAY　“T96CDE”，“T120ABC”，“T80FGA”

のような場合，有効なのはCH3の文字列中にある“T80”です．

（その他）

ミュージック制御命令は1小節を内部で128分割していました．サウンド制御命令では，1小節は192分割されますので，“L12”や“L24”などの音長も正確に表現できます．

また，音長の長さが255ステップをこえた場合は次のようになります．長い音符を表現する際には注意が必要です．

$$𝅗𝅥 = \frac{192}{2} = 96 \qquad \text{（2分音符）}$$

$$○. = \frac{192}{2} + 192 = 288 \text{（符点全音符）}$$

（288を256で割った余りの長さ）＝32

が実際に発音される音長の長さとなります．

エラーとはなりません．

(7) N₈₈-BASIC(86) によるプログラム例

(a) 演奏プログラムの例

N₈₈-BASIC(86) によるプログラムの例として“バッハのインベンション”を次に示します．

```
1000 '******************************
1010 '
1020 '             BACH
1030 '                   Invention
1040 '
1050 '******************************
1060 CLEAR ,&H9E00
1070 PLAY ALLOC 500,500,500
1080 PLAY "@46@V95MFZ0,3","@46@V95MFZ0,3","@46@V95MFZ0,3"
1090 '
1100 PLAY "T70"
1110 PLAY "L16RO5DC+DO4A8O5F8.EDEO4A8O5G8",,"O4L8DEFDABO5C+O4A"
1120 PLAY "L16RFEDA4RG+F+G+G+8.A","L16RO4AG+AE8O5C8.O4BABE8O5D8","L16O5D4RCO4BAO
5E4O4E4"
1130 PLAY "O5L16A4RDC+DG4RC+O4BO5C+","O3A4","L16RCO4BO5CF4RO4BABO5E4"
1140 PLAY "O4RDC+DO3A8O4F8.EDEO3A8O4G8","O5F4.A8G4RFEF","L16R8O4A8O5DEDCO4B8.O5C
+C+4"
1150 PLAY "RFEDB-4REDCA4","D4RB-AB-E4RAGA","RDC+DO4G4RO5CO4B-O5CO4F4"
1160 PLAY "RDCDO3G8O4F8.EDEC8B-8","D4RDEFG4RFED","RB-AO5CO4B-4RO5CO4B-AB-AGF"
1170 PLAY "RAGAL8FO5E-DCO4B-A","C2CO4B-AB-GO5DCD","E8B-8.AGAF2"
1180 PLAY "B-GO5CO4CL16RFEFC8A8","O4GB-AB-8O5CO4B-O5CO4A4R8O5F8","R4O4E4O3L8FGAF
"
1190 PLAY "RGFGC8B-8.O5FEFC8A8","RE8.R8G8.O4A8.R8O5F8","O4CDECFGAF"
1200 PLAY "RGFGC8B-8.O5CO4B-O5CO4F8O5F8","RE8.R8G8.R4..","O5CDECF4L16REDC"
1210 PLAY "RO4B-AB-E8O5E8.O4AGAD8O5D8","RDC+DG4RFEGF4","O4B-4RAB-GO5C+4DO4B-AB-"
1220 PLAY "R8C8.O4BABG+8A8.G+F+G","REDEO4A8O5D8.CO4BO5CO4F+8B8","L8G+AF+FE4D+D"
1230 PLAY "L8AD+EO3EL16O4RAGAE8O5C8","RAG+A8GF+GO3A4R4","L8C+CO3B4ABO4CO3A"
1240 PLAY "RO4BABE8O5D8.O4ED+EO3A8O4G8","R2O5RCO4BO5CE-4","O4EF+G+EA4L16RGF+G"
1250 PLAY "RF+EFD8O5C8.GF+GD8B-8","RD8.R8F+8G2","D8C8O3B-8A8GO4B-AB-O5CDCD"
1260 PLAY "RAGAD8O5C8.DCDO4G8O5F8","RF+B-AF4G","E-4RDE-CDO4B-AB-O5CO4B-AG"
1270 PLAY "REDEC8B-8.O4FEFC8A8","R2RAGB-AGFE","O5C8D8E8C8O4F4R8F8"
1280 PLAY "RGFGO3B-8O4G8.FEFO3A8O4F8","D4RFEDC+4RO4AO5DC","L8B-AGB-AGFD"
1290 PLAY "FEDFE-DC+DC+8D8E4","O4B-2RAG+BAGFG","GFGEABO5C+O4A"
1300 PLAY "RDC+DO3A8O4F8.EDEO3A8O4G8","F4RAGAB-8B8O5C8C+8","RO3AO4D4L16GFGEB-AG"
1310 PLAY "RF+EF+D8O5C8O4B-8A8B-AGA","D2.R8D8","ADO5C8.O4B-AB-AGF+GD8B-8"
1320 PLAY "RAGAD8G8.FEFO3B8O4E8","L8C+CO4BB-A4G+G","GF+EF+GFE-DC+8D8.CO3BO4C"
1330 PLAY "RDC+D8C+O3BO4C+","F+FE4D1.","D8O3G+8A4D1."
```

(b)　演奏に表情をつける

①の譜面は，表情を特につけずに演奏する例です．これをMMLで表すと例えば次のようになります．

```
PLAY "@2R8L16D8F+8D8A8D8>D8C+<BABAGF+GF+EL2DR4"
```

この例に次の様な表情をつけます．

Ⓐ　クレッシェンド
Ⓔ　デクレッシェンド
：徐々に音量が増す（減る）効果を付加します．実際には聴感上不自然でない程度に分割して音量を増減させます．

Ⓑ　スタカート：Qコマンドにより，音を短く切ります．

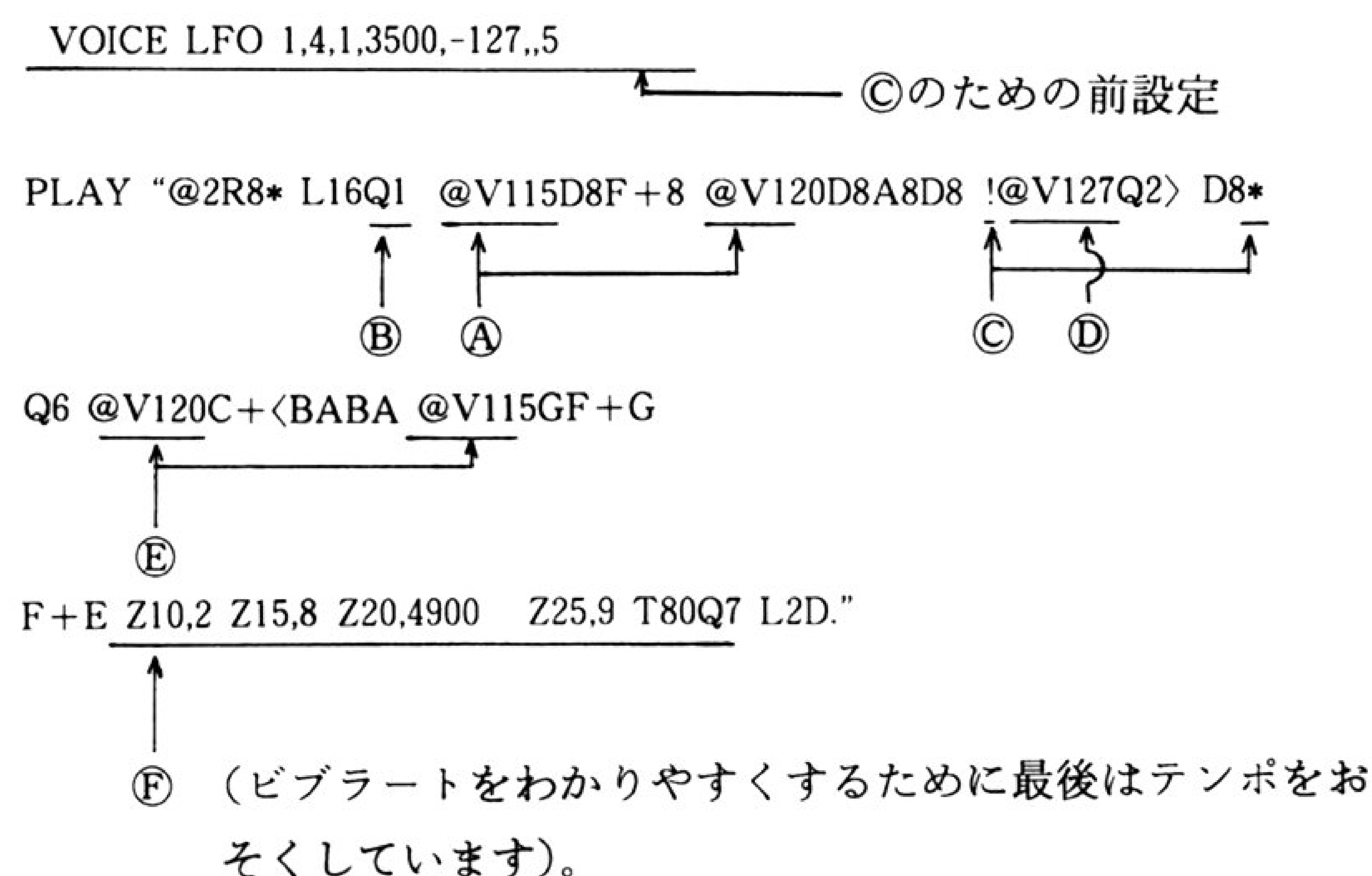
Ⓒ　ピッチベンド：LFO変調効果によりワンショット波形でピッチ変調効果を付加します．効果が不必要な部分では，＊コマンドで効果を停止させておき，！コマンドで，必要なところに来たとき効果を付加します．

Ⓓ　アクセント：アクセントは音量を一時的に増すことで表現します．

Ⓕ　ビブラート，トレモロ等：LFO変調効果のくり返し波形（三角波形）で，変調効果を付加します．演奏につれてLFO効果を変更する必要がある場合は，例題中のZコマンドは次のようなVOICE LFO文と同等です．

```
Z10,2  Z15,8  Z20,4900  Z25,9
  ||
VOICE LFO  x,2,8,4900,9
```

これらの表情を付加すると次の様になります．



(c) 基本音色を加工する

音色番号１の "BRASS2" の音色を利用して，犬の声を作ってみます．犬の声らしくするために，三角波ワンショット波形でピッチ変調をかけます．さらに，音程を低めに設定します．このように内蔵音色を加工して新しい音をつくることができます．

```
'
' BOW WOW
'
PLAY "@1":VOICE LFO 1,5,1,4500,127,,15
PLAY "O3L8Q4CR{CCC}4"
```

19.2.7 エラーメッセージ

サウンド拡張命令に関するエラーメッセージは次のとおりです．

○ "Sound buffer overflow" （エラー番号 132）

サウンドバッファ領域が足りない場合に発生します．

○ "Sound buffer not allocated" （エラー番号 133）

サウンドバッファ領域が確保されていないのに，拡張サウンド制御命令を使用しようとした場合に発生します．

○ "Out of octave" （エラー番号 134）

発生できる音域を超えた音を出そうとした場合（例えば "O9C+" や "O1C−" を指定した場合）に発生します．

19.2.8 外部オーディオ機器の使用

本体背面のサウンド用外部オーディオ機器接続端子を使用し，外部オーディオ機器を接続することにより，迫力あるサウンドが楽しめます．

オーディオ機器への接続にはミニプラグを使用します．ミニプラグを接続しますと，サウンド音は内部スピーカから出力されなくなりますが，BEEP音は内部スピーカから出力されます．なお，音声入力機能を持つアナログディスプレイが接続されている場合，この端子を使用しなくてもディスプレイ本体のスピーカより高出力のサウンドを楽しむことができます（この場合，内部スピーカからもサウンド音が出力されますので，内部スピーカのボリュームはしぼってください）．

第20章

ディップスイッチ

本体前面から操作出来る24個のディップスイッチがあります.

各スイッチの操作法と機能については「ハードウェアマニュアル」を参照してください.

第21章

メモリスイッチ

本体には不揮発性メモリが用意されており，この不揮発性メモリは電源が断たれても，メモリの状態を保持しています．ただし，電源を入れない状態で，2ヶ月間以上動作させないと不定になります．

注　意　次の場合には必ずディップスイッチを工場出荷時の状態にして電源投入状態を15時間以上続けてください．

(1)　長期間（2カ月間以上）電源を入れない状態があった場合．

(2)　本体を購入して初めて使用する場合（この場合長時間は不要です）

不揮発性メモリのバックアップ用電池の充電が行なわれ，メモリスイッチの状態をクリアし，システム既定値に設定します．

不揮発性メモリのメモリ状態を保持する特性を利用して，このメモリ上のそれぞれのビット状態「1」または「0」を，いわゆる「スイッチ」の「ON」「OFF」の状態に対応させたのが「メモリスイッチ」です．

21.1　メモリスイッチの使い方

論理スイッチ名	メモリ番地	データ（ビット位置）								機　　能	
		7	6	5	4	3	2	1	0		
									0	Xパラ	Xパラメータ無効
									1	メータ	Xパラメータ有効
								0		通信方式	全二重
								1			半二重
						1	0			データ	7ビット長
SW1	A3FE2					1	1			ビット長	8ビット長
					0					パリティ	なし
					1					チェック	あり
				0						パリティ	奇数パリティ
				1						指定	偶数パリティ
	システム	0	1							ストップ	1　ビット
	既定値	1	0							ビット長	1.5ビット
	$(48)_{16}$	1	1								2　ビット

論理スイッチ名	メモリ番地	データ（ビット位置）								機能	
		7	6	5	4	3	2	1	0		
SW2	A3FE6					0	0	0	0	ボーレート	無効
						0	0	0	1		75 ボー
						0	0	1	0		150 ボー
						0	0	1	1		300 ボー
						0	1	0	0		600 ボー
						0	1	0	1		1200ボー
						0	1	1	0		2400ボー
						0	1	1	1		4800ボー
						1	0	0	0		9600ボー
					0					日本語シフトコード	KI=$(1B4B)_{16}$ KO=$(1B48)_{16}$
					1						KI=$(1A70)_{16}$ KO=$(1A71)_{16}$
				0						C_R/C_R・L_Fコード受信時動作	$C_R(0D)_{16}$受信時：復帰＋改行
				1							C_R・$L_F(0D\ 0A)_{16}$受信時：復帰＋改行 $C_R(0D)_{16}$受信時：復帰
			0							RETURNキー押下時送信コード	$C_R(0D)_{16}$コード
			1								C_R・$L_F(0D\ 0A)_{16}$コード
	システム既定値	0								Sパラメータ	無効
	$(05)_{16}$	1									有効

論理スイッチ名	メモリ番地	データ（ビット位置）								機能	
		7	6	5	4	3	2	1	0		
SW3	A3FEA						0	0	0	メモリサイズ	128 Kバイト
							0	0	1		256 Kバイト
							0	1	0		384 Kバイト
							0	1	1		512 Kバイト
							1	0	0		640 Kバイト
						0				未使用	
					0					数値演算プロセッサ実装の有無	数値演算プロセッサ無し
					1						数値演算プロセッサ有り
				0						未使用	
			0							テキスト画面の初期カラー指定	白
			1								緑
		0								ターミナルモードでDELコード受信時動作	BS$(08)_{16}$コード扱い
		1									NUL$(00)_{16}$コード扱い
	システム既定値	0								入出力モードでDELコード受信時動作	DEL($(7F)_{16}$,$(FF)_{16}$)コード扱い
	$(04)_{16}$	1									NUL$(00)_{16}$コード扱い

論理スイッチ名	メモリ番地	データ（ビット位置）								機能		
		7	6	5	4	3	2	1	0			
SW4	A3FEE								0	0でなければならない(注2)		
								0		0でなければならない(注2)		
							0			拡張ROM接続 $(C8000)_{16}$〜$(C9FFF)_{16}$	なし	システム予約(注1)
							1				あり	
						0				拡張ROM接続 $(CC000)_{16}$〜$(CFFFF)_{16}$	サウンドボードなし	
						1					サウンドボードあり	
					0					拡張ROM接続 $(D0000)_{16}$〜$(D3FFF)_{16}$	RS−232C(第2回線,第3回線)ボードなし	
					1						RS−232C(第2回線,第3回線)ボードあり	
				0						拡張ROM接続 $(D4000)_{16}$〜$(D5FFF)_{16}$	GPIBインタフェースボードなし	
				1							GPIBインタフェースボードあり	
			0							拡張ROM接続 $(CA000)_{16}$〜$(CBFFF)_{16}$	なし	システム予約(注1)
	システム		1								あり	
	既定値	0								拡張ROM接続 $(CE000)_{16}$〜$(CFFFF)_{16}$	なし	
	$(08)_{16}$	1									あり	

注(1)：これらの拡張ROM空間は将来の機能拡張のために用意されているものです．絶対に使用しないでください．

注(2)：ゼロでない場合，システムの動作は保障されません．

論理スイッチ名	メモリ番地	データ（ビット位置）								機能	
		7	6	5	4	3	2	1	0		
SW5	A3FF2								0	PC-PR201系プリンタ使用の有無	使用しない
									1		使用する
								0		固定ディスクデバイス名優先指定使用	使用しない(フロッピィディスク→固定ディスクの順にディバイス名が割りふられる)
								1			使用する(固定ディスク→フロッピィディスクの順にデバイス名が割りふられる)
							0			固定ディスクユーザ識別名使用	使用する
							1				使用しない
						0				カラー画面ハードコピー/白黒画面ハードコピー	白黒画面ハードコピー 注(1)
						1					カラー画面ハードコピー 注(1)
		0	0	0	0					DISK BASIC立ち上げ時のBOOT装置の指定（これら以外の値が指定された場合ROM BASICが立ち上がる）	フロッピィディスク→固定ディスクの順にサーチする
		0	0	1	0						640KBフロッピィディスク装置のみをDISK-BASICの立ち上げ装置とする（他の装置は読みにいかない）
		0	1	0	0						1MBフロッピィディスク装置のみをDISK-BASICの立ち上げ装置とする（他の装置は読みにいかない）
		1	0	1	0						固定ディスク#1装置のみをDISK-BASICの立ち上げ装置とする（他の装置は読みにいかない）
	システム既定値 $(01)_{16}$	1	0	1	1						固定ディスク#2装置のみをDISK-BASICの立ち上げ装置とする（他の装置は読みにいかない）

注(1)：このスイッチはSW6.2^4ビットがONで，PC-PR201系カラープリンタが接続されている場合のみ意味を持ちます

論理スイッチ名	メモリ番地	データ（ビット位置）								機能	
		7	6	5	4	3	2	1	0		
SW6	A3FF6								0	未使用	
								0		未使用	
							0			未使用	
						0				モニタモード拡張機能使用の有無	使用しない
						1					使用する
					0					拡張画面ハードコピー機能（カラーコピーも可）使用の有無	使用しない
					1						使用する
	システム既定値 $(00)_{16}$			0						モデム-NCU内蔵電話制御機能使用の有無	使用しない
				1							使用する
			0							未使用	
		0									

21.2 メモリスイッチのセット

(1) システム既定値で使用する場合

・ディップスイッチSW 2のスイッチ5をOFFにしてください．

・リセットスイッチを押してシステムを立ち上げてください．

(2) システム既定値以外で使用する場合

・ディップスイッチSW 2のスイッチ5をONにしてください（下にたおします）．

・メモリスイッチを設定します．

メモリスイッチの設定方法には，専用のユーティリティを使用する方法とモニタモードを使用する方法の2つがあります．

通常メモリスイッチの設定は専用のユーティリティでおこないます．専用のユーティリティ(**switch.n88ユーティリティ**）に関しては，「第15章　システムディスクに格納されているユーティリティプログラム」を参照してください．

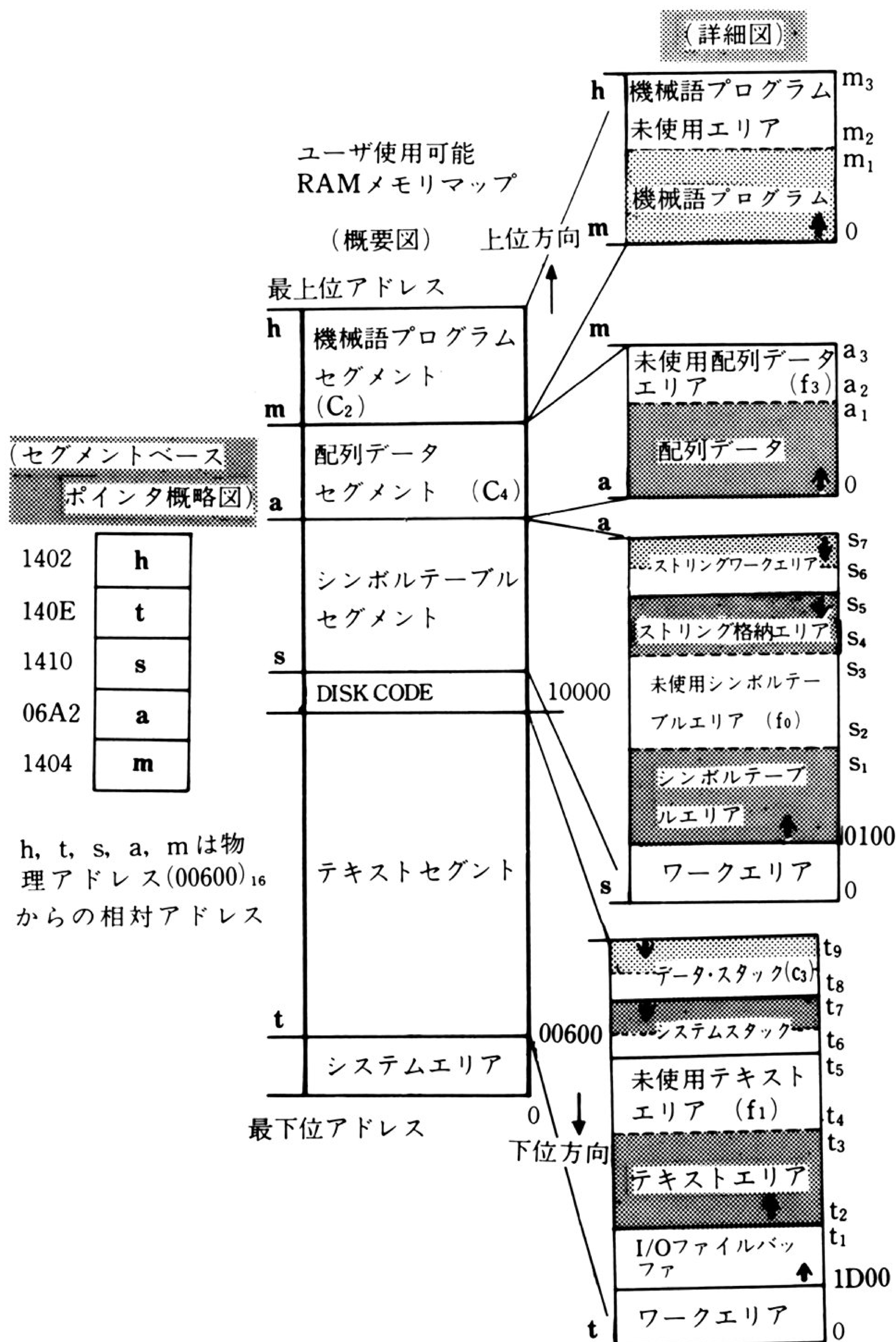
第22章

メモリマップ

22.1　システムのメモリ構成

システムのメモリ構成については「ハードウェアマニュアル」を参照してください.

22.2 N88-BASIC (86) のメモリ構成



1．セグメントベースポインタ概略図の説明

左端のアドレスは物理アドレス（00600）$_{16}$からの相対アドレスです．

1402：利用者RAM上限アドレスが格納されています．

140E：テキストセグメントベースアドレスが格納されています．

1410：シンボルテーブルセグメントベースアドレスが格納されています．

06A2：配列データセグメントベースアドレスが格納されています．

1404：機械語プログラムセグメントベースアドレスが格納されています．

各セグメントベースアドレスは物理アドレスの下位4ビットを除いた形で格納されています．

2．記号の説明

⬆または⬇：使用されてゆく方向を示しています．

▒：使用されているエリアを示しています．

fn：fre関数で知ることができる大きさ（たとえば，foは，FRE(0)で知ることができます）

cn：clear文により確保できる領域（たとえば，C_2はCLEAR文の第2パラメータで確保します）

概要図右側の数値：物理アドレスを16進数で示しています．

詳細図右側の数値：セグメント内アドレスを16進数で示しています．

3．メモリマップの説明

［　］は物理アドレス（16進数5桁で表わされます），(　)はセグメント内アドレス（オフセット）（16進数4桁で表わされます）を意味します．

① **メモリ全体は，システムエリアと4つのセグメントによって区分されています．**

② **［0～00600］システムエリアとして使用します．**割込ベクタ，入出力制御作業エリア等に使います．

③ **［00600～10000］「テキストセグメント」に使用します．**

・（0～1AFF）BASICインタプリタのワークエリアです．

・（1D00～t_1）How many files（0～15）？で応答した数値により定まるI/Oファイルバッファ領域あるいは接続されているディスク装置の管理情報領域，その他です．システム立ち上げ処理で確定します．

・（t_2～t_3）BASICプログラムが格納されます．下位アドレスから上位方向にエリアを使用してゆきます．

・（t_8～t_9）データスタック（演算スタックともよびます）として上位アドレスから下位方向にエリアを使用します（CLEAR命令によりサイズを変化させることができます）．

・（t_6～t_7）システムスタックとして上位アドレスから下位方向にエリアを使用します．

・（t_4～t_5）未使用テキストエリアです．fre(1)関数でこの大きさを知ることができます．

注　意　サウンド，GP-IB，RS-232C（第2回線/第3回線）等の拡張機能使用時はt_1～t_2の間をシステムで使用します．この分だけテキストエリアが少なくなります．

④ **［10000～S］DISK BASICモード時DISK CODE部がここにロードされます．**

DISK CODE部のサイズはシステムの起動形態によって異ります．

・標準DISK CODEのサイズは24KBです．

・通常は連文節変換入力機能が使用されますのでこれに90KBが加算されます．

単文節変換入力機能を選択した場合，90KBに変わり42KBが加算されます．

・拡張グラフィックモードでシステムが起動された場合（DIP SW 1の8番スイッチがONの場合），22KBが加算されます．

・拡張画面ハードコピー機能が選択された場合（メモリスイッチSW6・2^4ビットがONの場合），6KBが加算されます．

・モニタモードのディスクモード機能（たとえば，アセンブル/ディスアセンブル，ダンプディスク，エディットメモリ等の機能です）を使用する場合（メモリスイッチ6・2^3ビットがONの場合），11KBが加算されます．

・モデム-NCU内蔵電話制御機能を使用する場合（メモリスイッチ6・2^5ビットがONの場合），20KBが加算されます．

例

(1) 連文節変換入力機能を使用する場合
DISK CODEサイズ：114KB
（標準DISK CODE（24KB）＋連文節変換入力手続き（90KB））

(2) 単文節変換入力機能を使用する場合
DISK CODEサイズ：66KB
（標準DISK CODE（24KB）＋単文節変換入力手続き（42KB））

(3) 連文節変換入力機能に加え拡張グラフィックモードを使用する場合
DISK CODEサイズ：130KB
（標準DISK CODE（24KB）＋連文節変換入力手続き（90KB）＋拡張グラフィック手続き（22KB））

(4) 日本語入力機能は使用せず，拡張グラフィックモードを使用する場合
DISK CODEサイズ：40KB
（標準DISK CODE（24KB）＋拡張グラフィック手続き（22KB））

DISK CODE部はROM BASICモードの場合必要としません．[10000] からシンボルテーブルセグメントがとられます．

⑤ [S～a]「シンボルテーブルセグメント」に使用します．シンボルテーブルセグメント全体のサイズはCLEAR命令による配列データセグメントサイズの増減により変化します．

・（0100～s_1）シンボルテーブルエリアです．ラベル，変数名，関数名とその属性，数値型の変数データの値を格納するために，下位アドレスから，上位方向へむかって使用します．文字型変数のストリングディスクリプタもこのエリアに格納されます．

・（s_6～s_7）（0800＝2Kバイト）ストリングワークエリアです．文字列の演算のための作業エリアです．2Kバイトの領域です．

・（s_4～s_5）ストリング格納エリアです．上位アドレスから下位方向にとられます．文字型変数あるいは配列に代入される文字列はこのエリアに格納されます．

・（s_2～s_3）未使用シンボルテーブルエリアです．fre(0)関数でこの大きさを知ることができます．

注 意　シンボルテーブルセグメントの大きさは最大64Kバイトです．

⑥ [a～m]「配列データセグメント」に使用します．CLEAR命令によりサイズを変化させることができます．

・（0〜a_1）数値型の配列データと文字型配列のストリングディスクリプタが格納されます．下位アドレスから，上位方向へむかって使用します．

・（a_2〜a_3）未使用配列データエリアです．fre(3)関数でこの大きさを知ることができます．

注　意　1つの数値型配列変数の大きさは最大64Kバイトです．

⑦　[mo〜ho]「機械語プログラムセグメント」に使用します．

この領域は，BASICシステムによって干渉されません．機械語プログラムを展開するためのエリアです．

・このm番地は，clear命令第2パラメータによって定義します．下位4ビットを除いて指定します．

clear, m

・（0〜m_1）機械語プログラムで使用しているエリアです．下位アドレスから上位方向へ使用します．

・（m_1〜m_2）機械語プログラムの未使用エリアです．

注意1　メモリの使用状況はfre関数を使って知ることができます．

fre(0)＝未使用シンボルテーブルエリア（f_0）（s_2-s_3）

fre(1)＝未使用テキストエリア（f_1）（t_4-t_5）

fre(2)＝f_0+f_1

fre(3)＝未使用配列データエリア（f_3）（a_2-a_3）

注意2　メモリ領域の配分・確保のために，clear文を使います．メモリマップとclear命令のパラメータとの関係を示します．

clear [〈p_1〉], [〈p_2〉], [〈p_3〉], [〈p_4〉]

p_1　ダミーパラメータです．省略してかまいません．

p_2　BASICインタプリタが使用する上限値（m）

p_3　データスタックの大きさ（c_3）

p_4　配列データセグメントの大きさ（c_4）

第23章

N$_{88}$-BASIC(86)が扱うデータの内部形式

ここでは，BASICプログラム中の変数がメモリ上でどのように表現されているかを説明します．

(1) 変数の型

変数の型	数値部分のメモリ数	型宣言文[注1]	型宣言文字[注2]	変数名の例
整数型	2バイト	defint	%	AB%
単精度実数型	4バイト	defsng	!	AB, AB!
倍精度実数型	8バイト	defdbl	#	AB#
文字型	ストリングディスクリプタ 4バイト	defstr	$	AB$

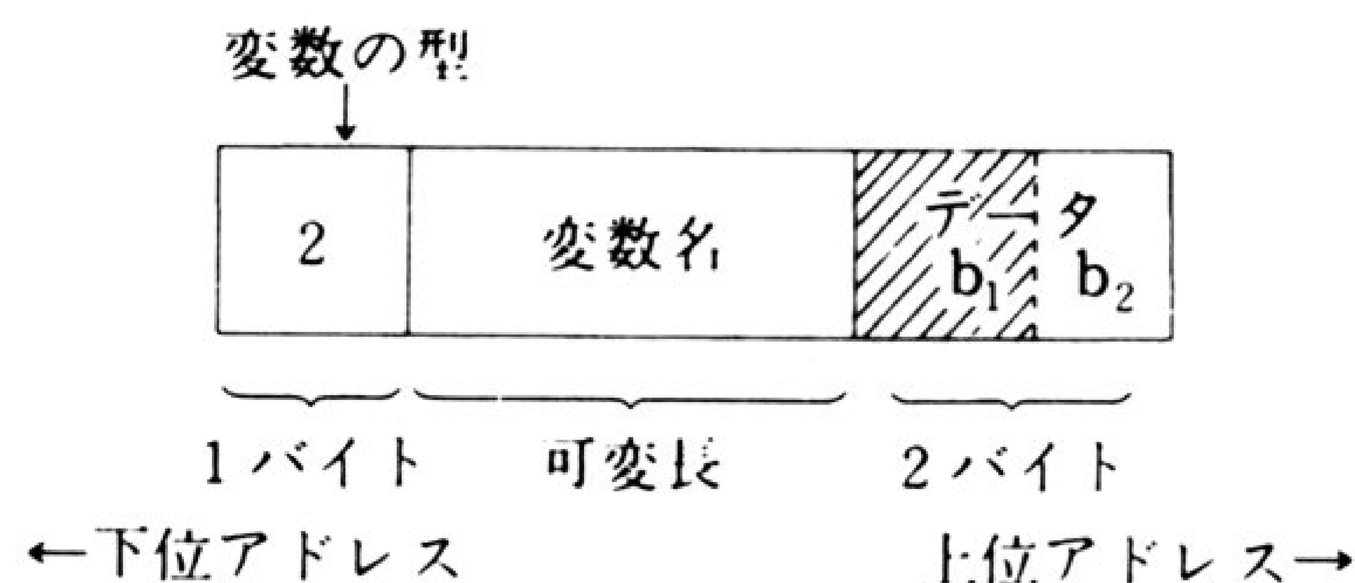
注1　型宣言文がないときはdefsngが実行されたものとして処理される．

注2　変数で型宣言をしないと型宣言文で指定した型になる．

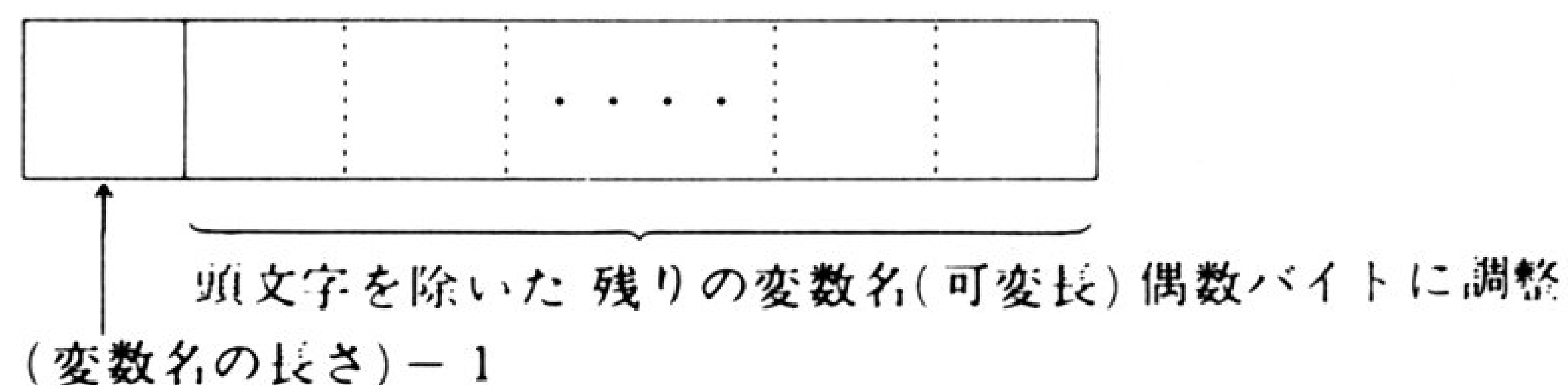
(2) メモリ表現

(a) 整数型変数

①メモリ表現形式



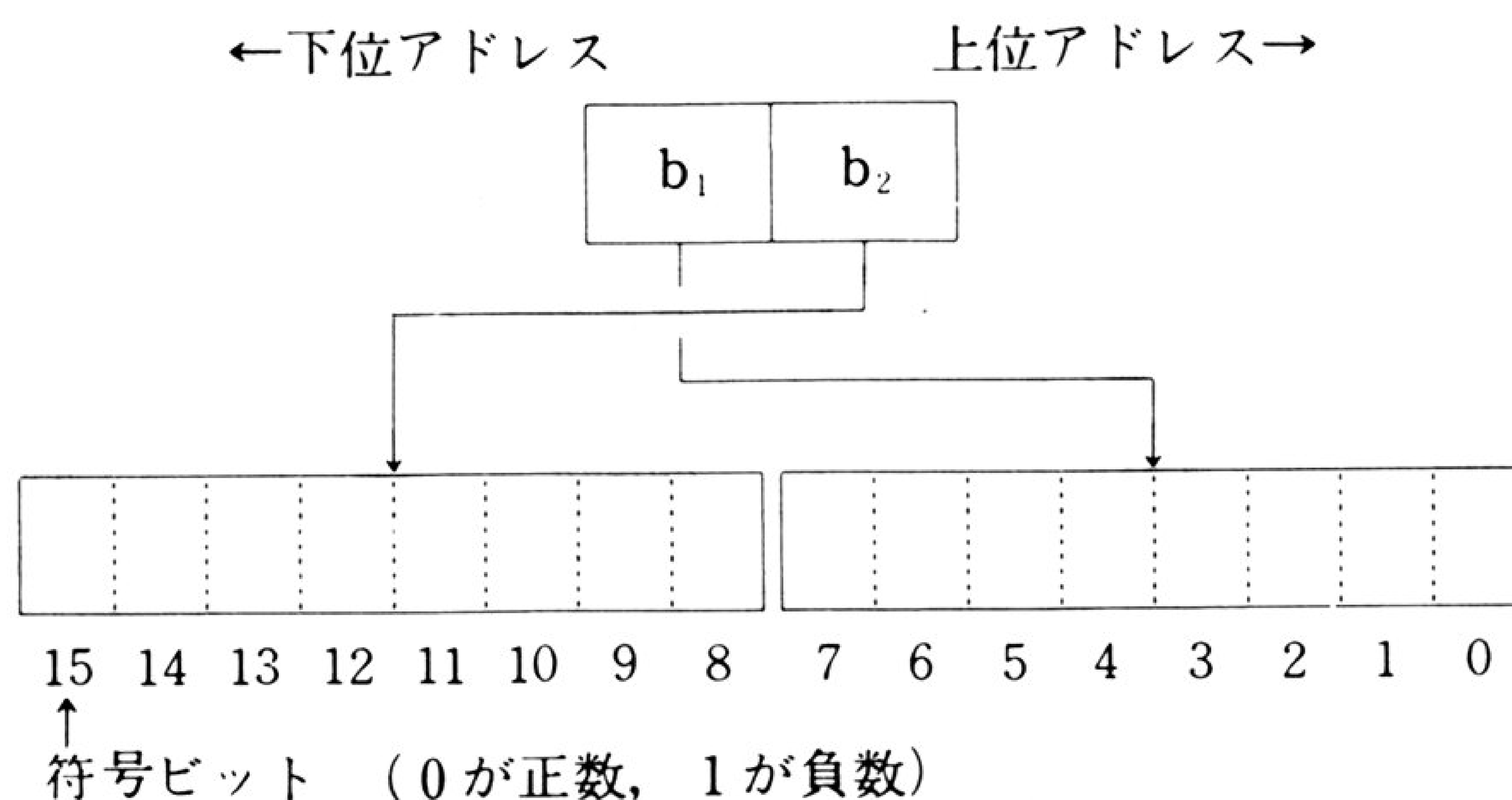
変数名
(すべての型に共通)



（例） 変数名 ABCDEFG%

```
06  42  43  44  45  46  47
↑   ↑
│   頭文字を除いた変数名.
│   文字数を偶数個に調整します.
変数名の文字数－1
```

②データ表現



整数の表現範囲　16進数　7FFF～8000（2の補数表現）

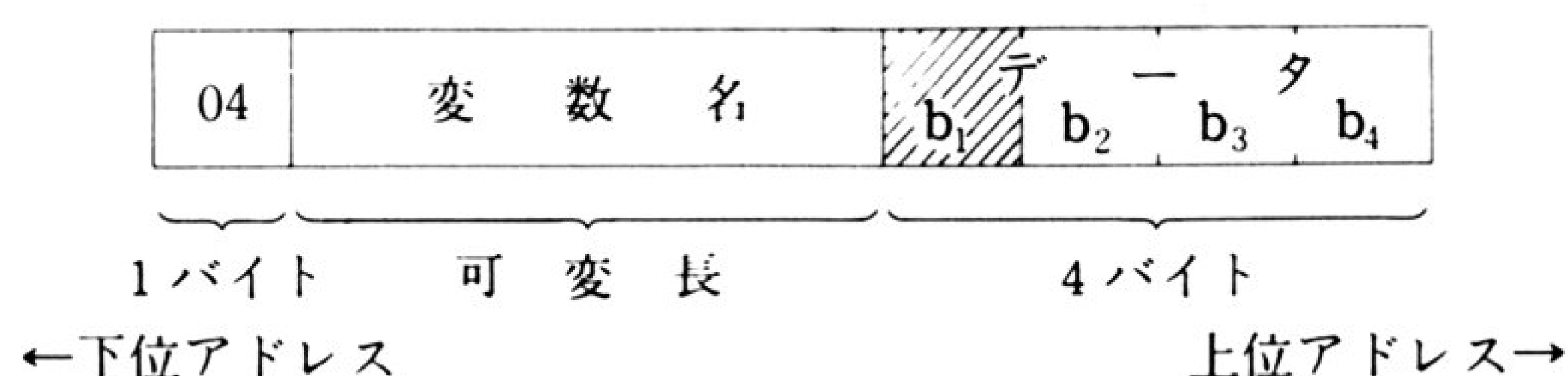
10進数　32767～－32768

注　意　2の補数表現で表わします.

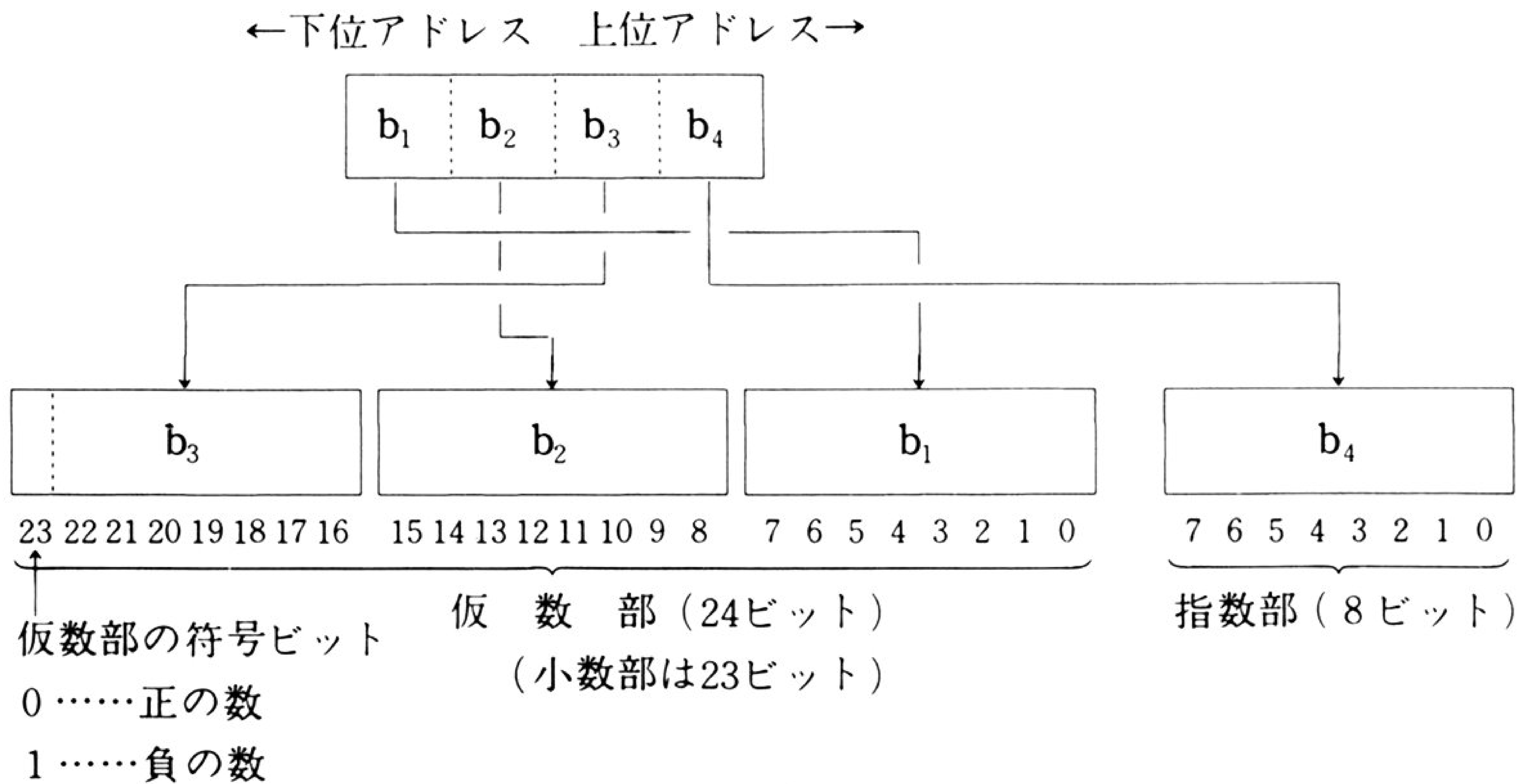
```
 ⋮          ⋮
0002  →    2
0001  →    1
0000  →    0
FFFF  →   －1
FFFE  →   －2
 ⋮          ⋮
```

(b)　単精度実数型変数

①メモリ表現形式



②データ表現



・仮数部は24ビットからなり，内1ビットは符号表示，残り23ビットが小数部を表わします．

・指数部は8ビットからなり，その表現形式は16進数で，$(81)_{16}$を0とし，$(82)_{16}$を1，$(80)_{16}$を－1として表現しています．$(FF)_{16}$は126，$(01)_{16}$は－128です．
$(00)_{16}$は仮数部の如何にかかわらず，この値を0とします．

FF	126
FE	125
⋮	⋮
82	1
81	0
80	－1
⋮	⋮
02	－127
01	－128
00	この数値を0にする．

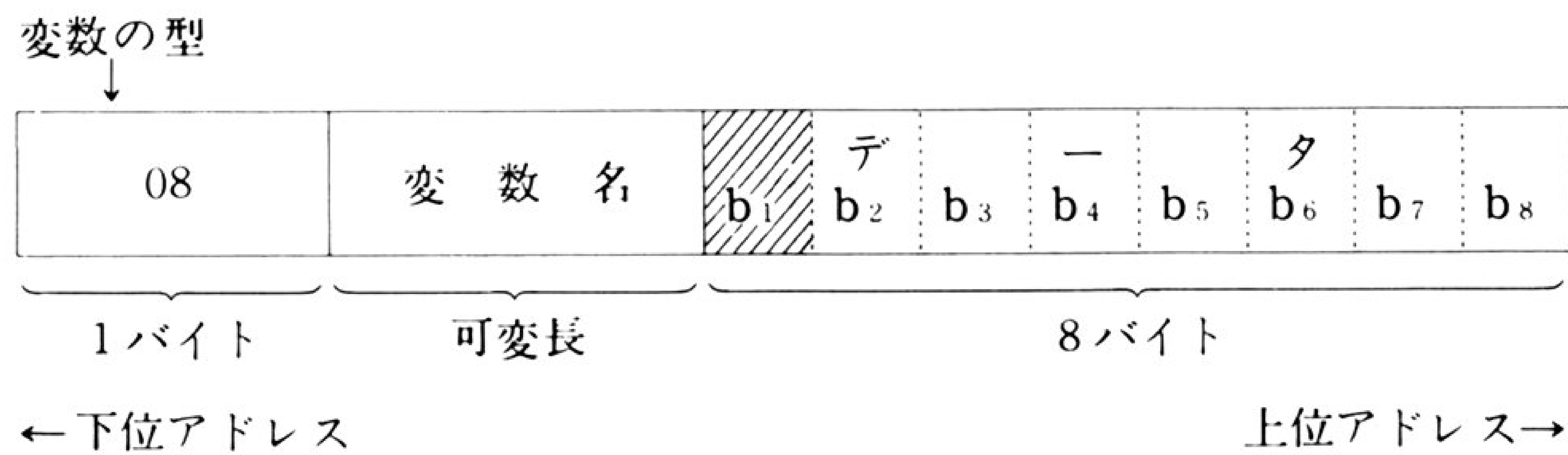
・絶対値の最大値，最小値は，次のようになります．

最大値＝$1.\underbrace{11\cdots11}_{23個}\times 2^{(FF)_{16}-(81)_{16}}$

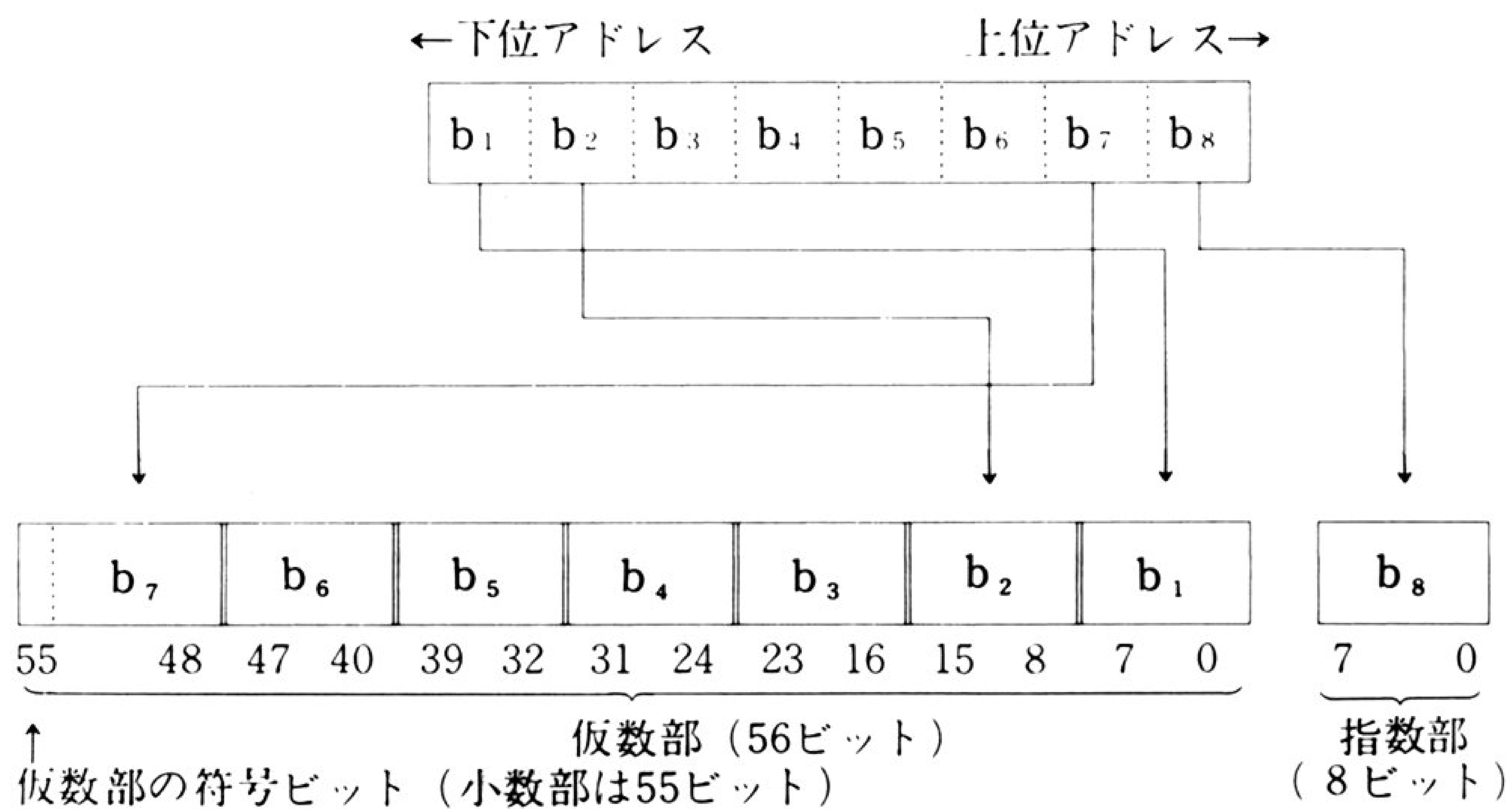
最小値＝$1.\underbrace{00\cdots00}_{23個}\times 2^{(01)_{16}-(81)_{16}}$

(c) 倍精度実数型変数

①メモリ表現形式



②データ表現



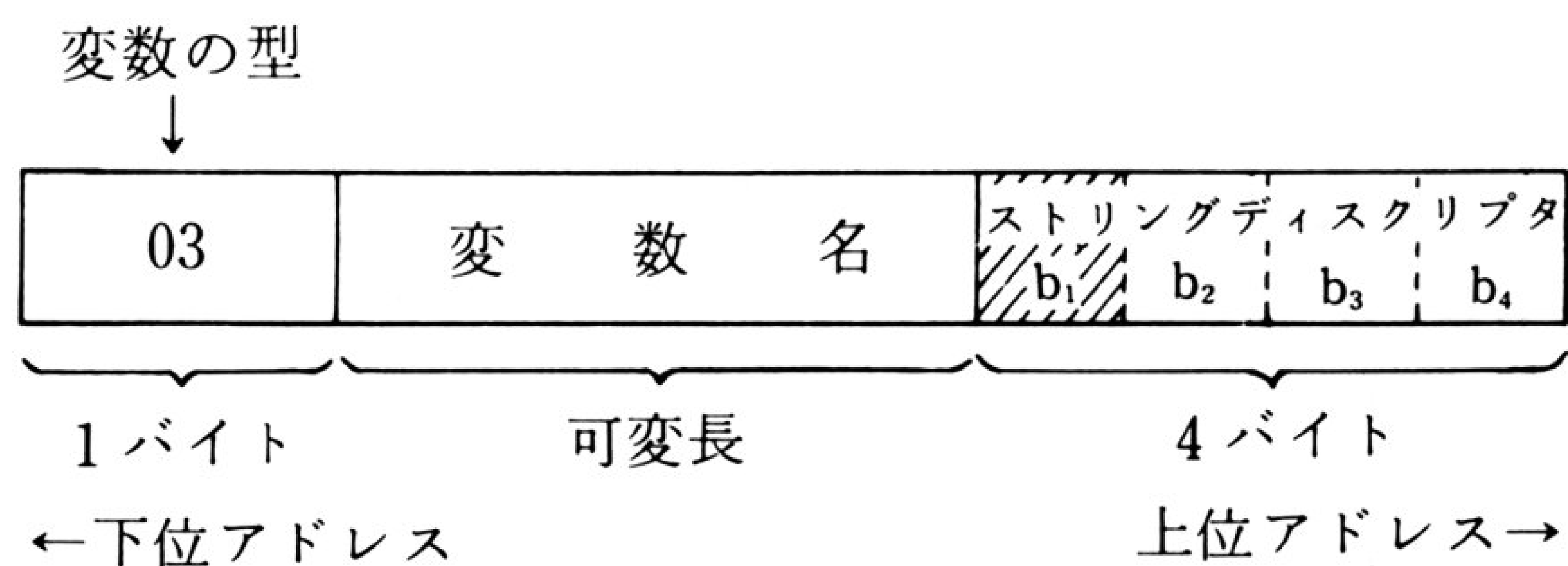
・仮数部は56ビットからなり，内1ビットは符号表示，残り55ビットが小数部を表わします．

・指数部は8ビットからなり，その表現形式は単精度実数型と同じです．

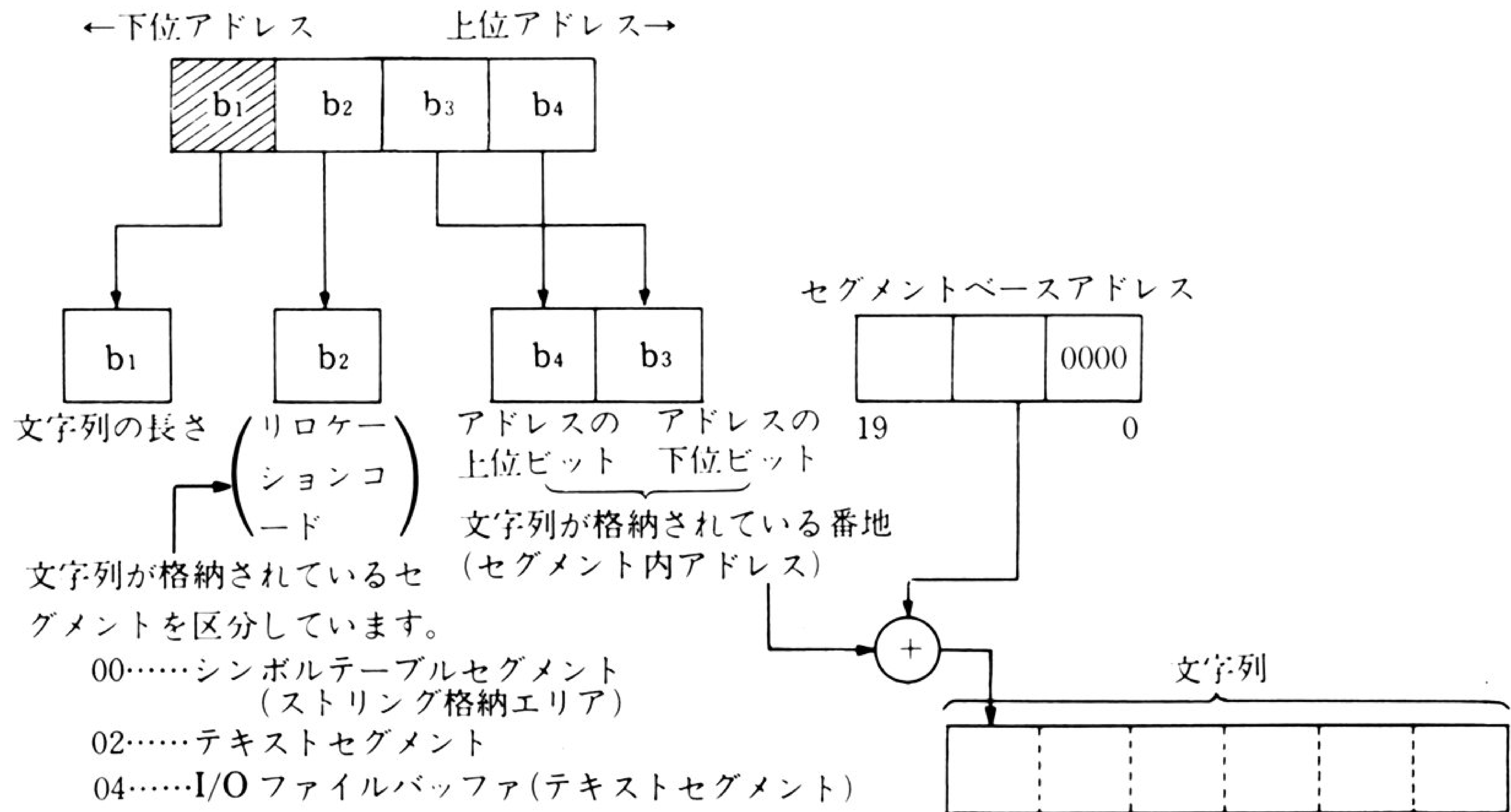
・絶対値の最大値，最小値についても，(b)と同様に表現できます．精度がよくなります．

(d) 文字型変数

①メモリ表現形式



②データ表現


←下位アドレス
上位アドレス→
b1
b2
b3
b4
b1
b2
b4
b3
セグメントベースアドレス
0000
19
0
文字列の長さ
リロケーションコード
アドレスの上位ビット
アドレスの下位ビット
文字列が格納されている番地（セグメント内アドレス）
文字列が格納されているセグメントを区分しています。
00……シンボルテーブルセグメント（ストリング格納エリア）
02……テキストセグメント
04……I/Oファイルバッファ(テキストセグメント)
+
文字列

付録．A　キャラクタコード表

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		D_E	SP	0	@	P		p			SP	ー	タ	ミ		
1	S_H	D_1	!	1	A	Q	a	q			。	ア	チ	ム		円
2	S_X	D_2	"	2	B	R	b	r			「	イ	ツ	メ		年
3	E_X	D_3	#	3	C	S	c	s			」	ウ	テ	モ		月
4	E_T	D_4	$	4	D	T	d	t			、	エ	ト	ヤ		日
5	E_Q	N_K	%	5	E	U	e	u			・	オ	ナ	ユ		時
6	A_K	S_N	&	6	F	V	f	v			ヲ	カ	ニ	ヨ		分
7	B_L	E_B	'	7	G	W	g	w			ァ	キ	ヌ	ラ		秒
8	B_S	C_N	(	8	H	X	h	x			ィ	ク	ネ	リ	♠	
9	H_T	E_M	)	9	I	Y	i	y			ゥ	ケ	ノ	ル	♥	
A	L_F	S_B	*	:	J	Z	j	z			ェ	コ	ハ	レ	♦	
B	H_M	E_C	+	;	K	[	k	{			ォ	サ	ヒ	ロ	♣	
C	C_L	→	,	<	L	¥	l	¦			ャ	シ	フ	ワ	●	
D	C_R	←	−	=	M	]	m	}			ュ	ス	ヘ	ン	○	
E	S_O	↑	.	>	N	^	n	~			ョ	セ	ホ	゛		
F	S_I	↓	/	?	O	_	o	DEL			ッ	ソ	マ	゜		DEL

注　意　SPは空白（スペース）コードを示します．

付録．B　日本語コード表（JIS第一水準および第二水準漢字コード表は「ハードウェアマニュアル」に記載されております）

文字列として扱える半角文字一覧表（テキスト画面に表示できる半角数字）

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0020	ⓈⓅ	!	"	#	$	%	&	'	(	)	*	+	,	−	.	/
0030	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0040	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0050	P	Q	R	S	T	U	V	W	X	Y	Z	[	¥	]	^	—
0060		a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0070	p	q	r	s	t	u	v	w	x	y	z	{	¦	}	~	
00A0	ⓈⓅ	。	「	」	、	・	ヲ	ァ	ィ	ゥ	ェ	ォ	ャ	ュ	ョ	ッ
00B0	ー	ア	イ	ウ	エ	オ	カ	キ	ク	ケ	コ	サ	シ	ス	セ	ソ
00C0	タ	チ	ツ	テ	ト	ナ	ニ	ヌ	ネ	ノ	ハ	ヒ	フ	ヘ	ホ	マ
00D0	ミ	ム	メ	モ	ヤ	ユ	ヨ	ラ	リ	ル	レ	ロ	ワ	ン	゛	゜

注　意　ⓈⓅは空白（スペース）コードを示します．

注　意　半角文字はプリンタで印字することはできません．

グラフィック画面に表示できる半角文字一覧表

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0020	Ⓢⓟ	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
0030	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0040	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0050	P	Q	R	S	T	U	V	W	X	Y	Z	[	¥	]	^	—
0060		a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0070	p	q	r	s	t	u	v	w	x	y	z	{	¦	}	~	
0080		。	「	」	、	・	を	ぁ	ぃ	ぅ	ぇ	ぉ	ゃ	ゅ	ょ	っ
0090	ー	あ	い	う	え	お	か	き	く	け	こ	さ	し	す	せ	そ
00A0	Ⓢⓟ	。	「	」		・	ヲ	ァ	ィ	ゥ	ェ	ォ	ャ	ュ	ョ	ッ
00B0	ー	ア	イ	ウ	エ	オ	カ	キ	ク	ケ	コ	サ	シ	ス	セ	ソ
00C0	タ	チ	ツ	テ	ト	ナ	ニ	ヌ	ネ	ノ	ハ	ヒ	フ	ヘ	ホ	マ
00D0	ミ	ム	メ	モ	ヤ	ユ	ヨ	ラ	リ	ル	レ	ロ	ワ	ン	゛	゜
00E0	た	ち	つ	て	と	な	に	ぬ	ね	の	は	ひ	ふ	へ	ほ	ま
00F0	み	む	め	も	や	ゆ	よ	ら	り	る	れ	ろ	わ	ん	゛	゜
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

注意 Ⓢⓟは空白（スペース）コードを示します.

グラフィック画面に表示できる1/4角文字一覧表

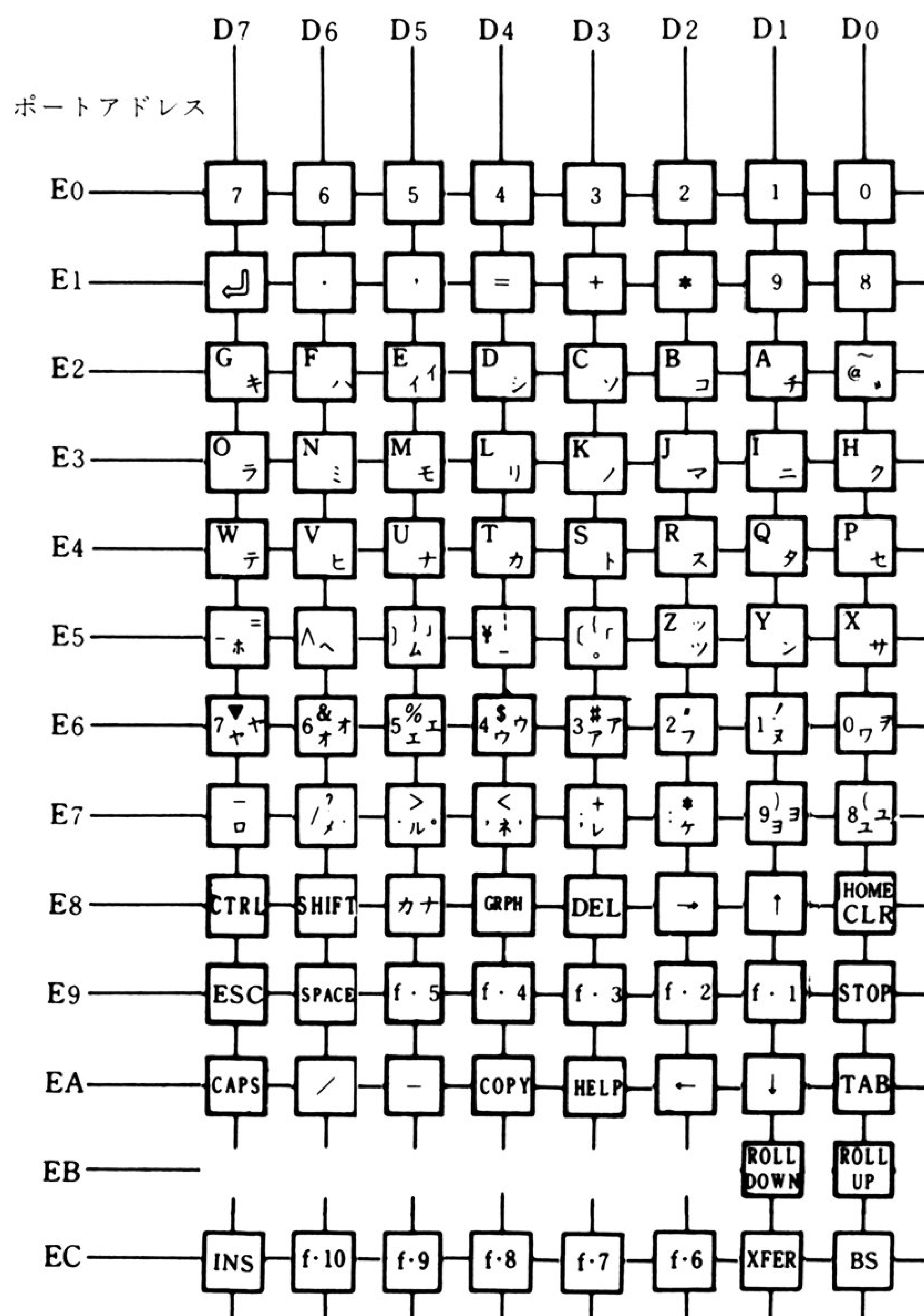
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0100		SH	SX	EX	ET	EQ	AK	BL	BS	HT	LF	HM	CL	CR	SO	SI
0110	DE	D1	D2	D3	D4	NK	SN	EB	CN	EM	SB	EC	→	←	↑	↓
0120	SP	!	"	#	$	%	&	'	(	)	*	+	,	−	.	/
0130	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0140	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0150	P	Q	R	S	T	U	V	W	X	Y	Z	[	¥	]	∧	—
0160		a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0170	p	q	r	s	t	u	v	w	x	y	z	{	¦	}	~	DEL
0180																+
0190													╭	╮	╰	╯
01A0	SP	。	「	」	、	・	ヲ	ァ	ィ	ゥ	ェ	ォ	ャ	ュ	ョ	ッ
01B0	ー	ア	イ	ウ	エ	オ	カ	キ	ク	ケ	コ	サ	シ	ス	セ	ソ
01C0	タ	チ	ツ	テ	ト	ナ	ニ	ヌ	ネ	ノ	ハ	ヒ	フ	ヘ	ホ	マ
01D0	ミ	ム	メ	モ	ヤ	ユ	ヨ	ラ	リ	ル	レ	ロ	ワ	ン	゛	゜
01E0	═	╞	╪	╡					♠	♥	◆	♣	●	○	╱	╲
01F0	×	円	年	月	日	時	分	秒								DEL
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

注 意 SPは空白（スペース）コードを示します.

付録．C　キーセンス

BASICのINP関数を使用し，キーボードのキーの押下状態をセンスすることができます．

キーセンスのためのポートアドレスはE0～ECで，ポートアドレスとキーの対応関係は次の通りです．



（例）　10 PRINT HEX$ (INP(&HE3))

```
10 PRINT HEX$ (INP(&HE3))
20 GO TO 10
```

を実行して「FF」が表示されたらH〜Oのキーは押されていません．もし「FE」が表示されていればHが，「F7」が表示されていればKが押されていることになります．

$(FF)_{16}=(11111111)_2$……押されていない

$(FE)_{16}=(11111110)_2$……Hのキーが押されている

$(FD)_{16}=(11111101)_2$……Ｉのキーが押されている

$(FB)_{16}=(11111011)_2$……J　〃

$(F7)_{16}=(11110111)_2$……K　〃

$(EF)_{16}=(11101111)_2$……L　〃

$(DF)_{16}=(11011111)_2$……M　〃

$(BF)_{16}=(10111111)_2$……N　〃

$(7F)_{16}=(01111111)_2$……O　〃

２個のキーが同時に押されているときは，上記以外の値になります．例えばＩとMが押されていれば，$(11011101)_2=(DD)_{16}$が得られます．

付録D　数値演算プロセッサの使用

数値演算プロセッサは浮動小数点データによる演算を高速化するための付加プロセッサです．

数値演算プロセッサが本体に実装されている場合，所定のメモリスイッチを設定することにより，BASICの数学関数の実行速度を速めることができます．

1．数値演算プロセッサの実装

「ハードウエアマニュアル」を御参照下さい．

2．メモリスイッチの設定

メモリスイッチ　SW3　2^4ビット：ON

メモリスイッチの設定は「switch.n88」ユーティリティを使用し簡単に行うことができます．「switch.n88」ユーティリティの使用方法については「第15章　システムディスクに格納されているユーティリティプログラム」を御参照下さい．

3．数値演算プロセッサの効果

BASICシステムにおいては次に示す数学関数あるいは演算処理が高速化されます．

SIN, COS, TAN, ATN, EXP, SQR, べき乗演算

四則演算は高速化の対象となりませんので御注意下さい．

索　引



PCH31
S9715111